KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

A Framework for Designing Decision Support Systems to Generate Artificial Intelligence Pipelines

Systems Design for AI Pipelines in Energy Time Series Applications

Giulia Rinaldi

Supervisor:
Prof. dr. ir. B. De Moor

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

June 2025

A Framework for Designing Decision Support Systems to Generate Artificial Intelligence Pipelines

Systems Design for AI Pipelines in Energy Time Series Applications

Giulia RINALDI

Examination committee:
Prof. dr. ir. J. Berlamont, chair
Prof. dr. ir. B. De Moor, supervisor
Prof. dr. ir. J. Suykens
Prof. dr. ir. M. F. Sien Moens
Dr. O. M. Agudelo
Prof. dr. L. Houthuys
(VUB AI-lab)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

June 2025

© 2025 KU Leuven – Faculty of Engineering Science
Uitgegeven in eigen beheer, Giulia Rinaldi, Kasteelpark Arenberg 10, 3001 Leuven, B-3001 Leuven (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotokopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm, electronic or any other means without written permission from the publisher.

Preface

> Stay afraid, but do it anyway. What's important is the action. You don't have to wait to be confident. Just do it and eventually the confidence will follow.
>
> *Carrie Fisher*

This quote has guided me throughout my research, reminding me that perseverance and action are more important than waiting for confidence to arrive. My Ph.D. journey has been a continuous battle, not only with academic challenges but also with self-doubt and fear. In these struggles, I found inspiration in Athena, the Greek goddess of wisdom and war, also known as Minerva.

Athena represents intelligence, strategy, and resilience, qualities that shaped my path. She is the inspiration behind the cover of this thesis: standing in the Colosseum, using a Decision Support System to determine the fate of gladiators, drawing upon her wisdom and judgment. The owl on her shoulder, a symbol of knowledge and insight, reminds us that clarity often comes through guidance and experience.

Her story resonates deeply with my own journey. Born from Zeus's mind, Athena embodies the balance between wisdom and strength, reason and courage. Like her, I have faced battles, not against external enemies, but against my own fears and uncertainties. There were moments when I doubted myself, when I felt lost, but like Athena's owl, I found clarity in the wisdom of those who supported me.

This Ph.D. has been a test of endurance, curiosity, and resilience. It has taught me that setbacks and failures are not obstacles but lessons, shaping the path

forward. Yet, no journey is walked alone. The strength I have found in myself is a reflection of those who have guided and supported me along the way.

Among them, my deepest gratitude goes to my supervisor, **Professor Bart De Moor**, for trusting me to be part of his team. He supported my research, gave me the space to grow and make mistakes, and guided me through the ups and downs of this journey, showing me both the right and wrong paths in research. Thank you for always being there. I always refer to him as "Professor Bart." While it is common for Ph.D. students to use only their supervisor's first name, I choose to include his title as a sign of my continued respect, in an informal yet appreciative way.

I am also deeply grateful to my jury members. **Professor Sien Moens** and **Professor Johan Suykens**, who have been part of my supervisory committee from the very beginning, provided invaluable insights and direction. Their expertise, guidance, and constructive feedback played a crucial role in shaping my research, and I sincerely thank them for their persistent support.

A warm thank you to **Professor Berlamont**, the chairman of my defense, for showing great interest in my research and making this final step of my Ph.D. journey a meaningful and stimulating discussion. I am also honored to have **Professor Houthuys** as part of my jury. She immediately accepted my invitation, and her enthusiasm truly touched me. I met her briefly at an event, and her kindness and optimism left a lasting impression. I am grateful for her presence and valuable insights.

A special thank you to **Dr. Agudelo**, whom I always call Mauricio because he holds a special place in my heart. He and **Willem** were my safety net, giving me the confidence and encouragement I needed throughout my Ph.D. Without them, this journey would have been much harder. They are brilliant professionals and even better friends. Having Mauricio on my jury is the cherry on top, and I couldn't be happier to have met them.

I also extend my heartfelt gratitude to **Willem's family**: Kirsten, Sophia, Alexia, Aurelio, and Emilio. Every time I see them, I feel like part of their family. Even though I still don't speak Dutch, the children always involve me in their games, filling our time together with joy and laughter. Thank you for the warmth, fun, and light moments.

At the university, I was fortunate to meet incredible people, especially my strong, beautiful, and intelligent women at Stadius: **Melanie, Lola, Naomi, Zahra, Yamini, and Wanqiu**. We shared smiles and tears, worries and successes, failures and victories. We stood by each other, and I know we will continue to do so beyond Ph.D. life. This support was essential in believing in ourselves and sharing our experiences. Thank you for your love and encouragement—and

for the many more celebrations yet to come.

The Ph.D. also allowed me to meet remarkable professionals who, with their kindness and efficiency, made our lives easier. A special thank you to **Ida**, who handled our endless requests and last-minute changes with patience. **Aldona** was always ready to listen and offer her wisdom, making even the most difficult days feel manageable. **Elsy, John, and Sofie** were always there to provide guidance, making complex processes simple and efficient.

Many thanks to the people I was lucky enough to call colleagues. **Katrien**, first and foremost, you always had a smile and a kind word for me. You are an impressive professional—strong, serious, and determined. Even for a short period, I had the pleasure of meeting **Philippe**, a kind and fantastic person, both personally and professionally. He offered valuable advice, and I remember him with deep appreciation and respect.

I am also grateful to the people who helped me during my first weeks in Belgium: **Christof, Oliver, Bram, and Bob**. You welcomed me when I knew nothing, showed me around, and made my transition so much easier. Later, my journey was enriched by the presence of **Vincent, Thomas, Lukas, Sibrien, Sartak, Hans, Sem, Jaldert, Yoeri, Alexandre, and Axel**, who completed the group and made the experience even more memorable.

A special thank you to **Kostas**. We worked together, and thanks to his contributions, I was able to publish my idea. He inspired me, and I hope he is as happy with our collaboration as I am.

I would also like to thank the **Flanders AI Program** and the incredible people behind it, especially **Sabine and Lorenza**, for their dedication to making our research visible and continuously stimulating new ideas. Thanks to this program, I met amazing people, and my gratitude to them is boundless.

Ph.D. life is never easy, and at one point, I went through an incredibly difficult and dark period. **Aras, Koen, and Thijs**, but especially **Jonas**, helped me regain confidence, continue my work, and ultimately publish my first paper. They cannot imagine how much their support meant to me, and I will always be grateful to them.

The Ph.D. journey is unique because, at some point, you realize that your most demanding, critical, and persistent colleague is... yourself. Through this process, you don't just learn about research—you end up learning what you actually want (and don't want) in life. I feel incredibly lucky that, at this very moment, I'm working in the best place I could have imagined. And this is not just because of the work itself, but because of the amazing people around me.

Jonathan, my new manager, gave me the same trust and support I had the

privilege to receive from Professor Bart. His empathy and patience gave me the space to grow—both technically and personally—and for that, I am deeply grateful.

A huge thank you to **Engin**, the first person who welcomed me and with whom I now work daily. I keep learning so much from him—professionally, but also through his calm and grounded perspective. I'm truly grateful to be walking this path with him.

To **Alexandra, Edouard, Leonor, and Joren**, thank you for making me feel part of the team and part of something meaningful from day one. You answered all my (sometimes embarrassingly basic) questions, supported me during my preliminary defense, and survived all my last-minute panic moments with impressive grace and kindness. You have always had a smile and an encouraging word for me, and I will never forget it.

A special thank you to **Benoit**, my other side of the medal at work. A wonderful professional and an even better person. I love our brainstorming sessions and interesting debates. We're figuring things out together, and that, in itself, is a gift.

And yes, at work there is also a little corner of Italy. Thank you **Francesco and Etienne** for being there with a smile, a coffee, and a Cantuccio (or two). Your presence makes even the toughest days feel a bit sweeter.

Last but certainly not least, I want to express my immense gratitude to **Richard**, my English teacher. I cannot count how many times he read my thesis, patiently correcting and refining my words. He listened to my frustrations, encouraged me to persevere, and played a crucial role in helping me reach the finish line. He is an exceptional professional, and I feel so lucky that he is now part of my life.

Tutto questo non sarebbe stato possibile senza l'amore e il sostegno incondizionato della mia mamma e del mio papà. Mi hanno sempre incoraggiata a seguire la mia strada, sostenendomi con fiducia, pazienza, forza, e anche con il supporto pratico necessario per affrontare questo percorso. Non mi hanno mai lasciata sola, nemmeno nei momenti più difficili. Spero, con tutto il cuore, di essere riuscita a renderli orgogliosi.

Un ringraziamento speciale va a **Daniela e Massimo**, i meravigliosi compagni di vita dei miei genitori. Con affetto e comprensione, sono stati loro accanto ogni giorno, affrontando insieme ansie, emozioni e piccole follie quotidiane. Grazie per essere parte di questa famiglia allargata e preziosa.

Un grazie di cuore a tutta la mia famiglia: **zia Mariana, zia Paola, zia Lorella, zia Anna, zio Andrea, zio Agostino, Mauro, Martina, Silvia,**

Mattia, Alessandro, Giampiero e ai cugini acquisiti **Carla, Marco e Diego**, che non mi hanno mai fatto mancare affetto e incoraggiamento, anche a distanza.

Un grazie speciale a **zia Antonella**, che mi ha sempre trattata come una figlia. La dedizione e l'amore che dimostra verso di me e verso la mamma sono un dono prezioso, silenzioso e imprescindibile.

Alle mie nonne, **nonna Gianna**, che non c'è più ma che porto sempre nel cuore, e **nonna Rosanna**, che con la loro forza, tenerezza e saggezza sono le mie radici. Con loro sono cresciuta: loro sono la mia storia, e senza di loro non sarei quella che sono oggi.

Grazie anche alla mia "famiglia acquisita", in particolare a **Federico**, a cui voglio un bene infinito, e a **Claudio e Roberta**, che mi hanno sempre trattata come una nipote e non hanno mai mancato un momento importante della mia vita. Il loro affetto costante è stato un punto fermo per me. Un grazie speciale a **Martina**, a cui auguro di brillare sempre, e a **Lorenzo**, che prima o poi convincerò ad assumermi come pet-cuddler... o almeno a regalarmi un gattino.

Un pensiero speciale va alle mie amiche del cuore: **Jessica e Alessandra**. Più che amiche, sorelle. Con la loro saggezza, ironia e amore mi hanno sempre dato il consiglio giusto, anche quando non volevo ascoltarlo. A **Melissa**, amica d'infanzia, sorella acquisita, che riesce sempre a riportarmi alla calma e all'essenziale. E ai miei migliori amici, **Dj, Andrea e Claudio**, con cui posso essere me stessa, ridere a crepapelle e non sentirmi mai giudicata. Voi tutti siete la mia libertà, il mio rifugio, il mio affetto più autentico.

Durante gli anni della triennale ho vissuto uno dei periodi più belli e intensi della mia vita, grazie a un gruppo di amici straordinari che mi ha fatto sentire imbattibile: **Lorenzo, Angelo, Serena, Andrea, Carmela, Angela, Luciana, Francesco, Jacopo, Enzo, Gianmarco, Noemi e Simona**. Anche se le nostre strade si sono fisicamente divise, la connessione rimane forte. Sappiamo che ci saremo sempre gli uni per gli altri. Grazie di cuore!

Un grazie speciale a **Federica**. Con delicatezza e dedizione mi ha aiutata a mettere ordine nei pensieri, a scoprire nuove parti di me, e ad affrontare momenti difficili con coraggio e semplicità. Le sarò sempre grata.

Un agradecimiento entrañable también a los padres, **Marta y Alejandro**. Me aceptaron desde el primer momento como parte de ellos, haciéndome sentir siempre bienvenida, cuidada y querida. Gracias por su cariño, por los momentos compartidos y por ser un nuevo pilar en esta etapa de mi vida.

Agradezco de corazón a toda la familia de Fernando,**tías, tíos, primos y primas**, por abrirme las puertas con tanta calidez. Cada encuentro, cada

palabra de aliento y cada gesto de afecto ha contribuido a que me sienta en casa, incluso estando lejos de la mía. Gracias por acompañarme con alegría, por su generosidad y por brindarme una nueva familia en la que me siento afortunada de estar.

A **los amigos de Fernando**, gracias por haberme recibido con tanto cariño y por incluirme en su grupo como si siempre hubiera estado allí. Gracias también por ayudarme (o al menos intentarlo) a soportar a Fernando en su locura diaria —aunque a veces parezca que lo alientan más de lo que lo calman.

Dulcis in fundo, un grazie profondo al mio compagno di vita e di avventure. L'uomo che sceglie ogni giorno di restarmi accanto, anche dopo aver scoperto i miei difetti. A lui devo tanto. Insieme stiamo costruendo una vita fatta di sogni, ironia e piccole grandi soddisfazioni. La nostra famiglia, **Chipi, Mimi, Felix e Lulù Stella** compresi, è il sogno realizzato di una bambina che credeva ancora nel principe azzurro. **Fernando** è il mio principe. Ogni giorno conquista il mio cuore e mi fa sentire la donna più fortunata del mondo. Gracias, mi amor. *Somos tal para cual.*

I am a lucky person. During this journey, I have felt alone, isolated, and hopeless. But the people I mentioned above helped me get through the darkest moments. They showed me the bright side, especially when I couldn't see it.

To conclude, I want to dedicate this thesis to myself. It might sound cheesy, but overcoming all those challenges has shown me my strength and the fact that I can do it.

Thank you.

Abstract

The rapid rise of Artificial Intelligence (AI)-based systems have transformed various domains, but their accessibility, reliability, and transparency remain major concerns. Despite the widespread adoption of AI, many existing frameworks fail to balance usability for non-experts with the technical depth required by developers. Open-source solutions often lack robustness, while proprietary tools restrict adaptability and limit trust in AI-driven decision-making.

This dissertation addresses this gap by developing frameworks that enhance Decision Support Systems (DSS) by integrating AI pipelines, focusing on time series applications in the energy domain. The research aims to improve interactions between humans and AI systems by making them more trustworthy, accessible, and user-friendly for both experts and non-experts. To achieve this, three interconnected frameworks were developed: a Data Quality Module (DQM), the DSS4EX framework, and the comprehensive EX-DSS system.

The DQM Framework introduces a systematic approach to assessing and improving data quality during the preprocessing phase, using metrics aligned with the European Union guidelines. It computes indicators for data completeness, consistency, and cleaning effort, providing actionable insights for users to enhance dataset reliability. The module's effectiveness was validated using energy time series data, demonstrating its ability to streamline data cleaning and support informed decision-making.

Designed to make AI pipelines more comprehensible, DSS4EX enables users to explore and configure AI models through an intuitive interface. It bridges the gap between complex AI methodologies and user understanding by incorporating guided workflows, expert recommendations, and modularity. DSS4EX was tested through an application in electricity demand forecasting, which proves its capability to support the decision-making process and foster broader AI adoption.

EX-DSS integrates the DQM and DSS4EX frameworks into a comprehensive system, supporting the full lifecycle of AI pipeline design and deployment. Having been assessed through a software demo to forecast energy consumption in commercial buildings, the EX-DSS framework showed to enhance user collaboration, data reproducibility, and modularity, which proved to be effective in improving energy efficiency. The inclusion of project-specific recommendations and human-in-the-loop features ensured transparency and adaptability for users with diverse expertise levels.

The proposed frameworks demonstrated significant real-world applicability, offering solutions to improve operational efficiency, data-driven decision-making, and user engagement. As a result, this research has contributed to the democratization of AI technologies, fostering their adoption in critical domains such as energy, healthcare, and finance. Future research will focus on expanding these frameworks' capabilities for real-time data processing, privacy-preserving mechanisms, and educational tools, enabling broader interdisciplinary applications and innovations.

Beknopte samenvatting

De snelle opkomst van op Kunstmatige Intelligentie (AI) gebaseerde systemen heeft verschillende domeinen getransformeerd, maar hun toegankelijkheid, betrouwbaarheid en transparantie blijven grote aandachtspunten. Ondanks de brede adoptie van AI slagen veel bestaande frameworks er niet in om gebruiksvriendelijkheid voor niet-experts in evenwicht te brengen met de technische diepgang die ontwikkelaars nodig hebben. Open-source oplossingen missen vaak robuustheid, terwijl propriëtaire tools de aanpasbaarheid beperken en het vertrouwen in AI-gestuurde besluitvorming verminderen.

Deze dissertatie pakt deze kloof aan door frameworks te ontwikkelen die Decision Support Systems (DSS) verbeteren door AI-pijplijnen te integreren, met een specifieke focus op tijdreeksapplicaties in de energiesector. Het onderzoek beoogt de interactie tussen mensen en AI-systemen te verbeteren door deze betrouwbaarder, toegankelijker en gebruiksvriendelijker te maken voor zowel experts als niet-experts. Om dit te realiseren, werden drie onderling verbonden frameworks ontwikkeld: een Data Quality Module (DQM), het DSS4EX-framework en het uitgebreide EX-DSS-systeem.

Het DQM-framework introduceert een systematische aanpak voor het beoordelen en verbeteren van de datakwaliteit tijdens de preprocessingsfase, waarbij metrieken worden gebruikt die in lijn zijn met de richtlijnen van de Europese Unie. Het berekent indicatoren voor datavolledigheid, consistentie en schoonmaakinspanning, en biedt bruikbare inzichten voor gebruikers om de betrouwbaarheid van datasets te verbeteren. De effectiviteit van de module werd gevalideerd met behulp van tijdreeksgegevens uit de energiesector, wat de mogelijkheid van het systeem aantoonde om gegevensreiniging te stroomlijnen en geïnformeerde besluitvorming te ondersteunen.

DSS4EX is ontworpen om AI-pijplijnen begrijpelijker te maken en stelt gebruikers in staat om AI-modellen te verkennen en configureren via een intuïtieve interface. Het overbrugt de kloof tussen complexe AI-methodologieën

en gebruikersbegrip door begeleide workflows, deskundige aanbevelingen en modulariteit te integreren. DSS4EX werd getest in een toepassing voor de voorspelling van elektriciteitsvraag, wat de effectiviteit ervan aantoonde in het ondersteunen van besluitvormingsprocessen en het bevorderen van bredere AI-adoptie.

EX-DSS integreert de DQM- en DSS4EX-frameworks in een uitgebreid systeem dat de volledige levenscyclus van AI-pijplijnontwerp en implementatie ondersteunt. Dit systeem werd geëvalueerd via een softwaredemonstratie voor het voorspellen van energieverbruik in commerciële gebouwen. Het EX-DSS-framework bleek samenwerking tussen gebruikers, reproduceerbaarheid van data en modulariteit te verbeteren, wat effectief was in het verhogen van energie-efficiëntie. De opname van projectspecifieke aanbevelingen en human-in-the-loop-functionaliteiten zorgde voor transparantie en aanpasbaarheid voor gebruikers met verschillende expertiseniveaus.

De voorgestelde frameworks hebben een significante praktische toepasbaarheid aangetoond en bieden oplossingen om operationele efficiëntie, datagestuurde besluitvorming en gebruikersbetrokkenheid te verbeteren. Als gevolg hiervan heeft dit onderzoek bijgedragen aan de democratisering van AI-technologieën en de adoptie ervan in cruciale domeinen zoals energie, gezondheidszorg en financiën bevorderd. Toekomstig onderzoek zal zich richten op de uitbreiding van de mogelijkheden van deze frameworks voor real-time gegevensverwerking, privacybeschermende mechanismen en educatieve hulpmiddelen, waardoor bredere interdisciplinaire toepassingen en innovaties mogelijk worden.

Contents

Abstract vii

Beknopte samenvatting ix

Contents xi

List of Figures xv

List of Tables xxiii

1 Introduction 1
- 1.1 Overview . . . 1
- 1.2 Context . . . 2
  - 1.2.1 Introduction to Decision Support System . . . 3
  - 1.2.2 DSS and Importance of Data Quality . . . 9
  - 1.2.3 DSS and Trustworthiness of Intelligent Systems . . . 10
  - 1.2.4 DSS and Artificial Intelligence Pipelines . . . 11
  - 1.2.5 DSS and Current Technologies . . . 13
- 1.3 Research Objectives . . . 15
- 1.4 Methodology . . . 16
- 1.5 Thesis Contributions . . . 17
- 1.6 Thesis Scope and Limitations . . . 20
- 1.7 Structure Thesis . . . 21
- 1.8 Acknowledgments . . . 23

2 DQM: Data Quality Module for Decision Support System 25
- 2.1 Introduction . . . 27
- 2.2 Related Work . . . 28
- 2.3 Chapter Objectives and Challenges . . . 30
- 2.4 The DQM Framework . . . 32
- 2.5 An industrial application of DQM at a proof-of-concept level. . . 36

2.5.1 General Description 38
2.5.2 DQM Real-life implementation at a proof-of-concept level 43
2.5.3 Implementation Details 55
2.6 Conclusions 56

3 DSS4EX: Framework to explore Artificial Intelligence Pipelines 59
3.1 Introduction 61
3.2 Literature Review 63
3.2.1 Software Tools for AI Pipeline Training 63
3.2.2 Software Tools for Public Understanding of AI 64
3.2.3 Decision Support System Framework 66
3.3 Chapter Objectives and Challenges 67
3.4 The DSS4EX Framework 68
3.5 An industrial application of the framework as software demo: DSS4EX for the Decomposition Residuals Deep Neural Network pipeline 72
3.5.1 General Description 72
3.5.2 DSS4EX Real-Life implementation as software demo . . 77
3.5.3 Software Demo Description 81
3.5.4 Implementation Details 94
3.6 Results and Discussions 100
3.7 Conclusions and Future Works 101

4 EX-DSS: An Explorative Decision Support System 103
4.1 Introduction 105
4.2 Background 106
4.3 Chapter Objectives and Challenges 107
4.4 EX-DSS framework 108
4.5 An industrial application of the framework as software prototype: EX-DSS for Smart Plug Forecasting Pipelines 110
4.5.1 General Description 111
4.5.2 EX-DSS Real-Life implementation as software prototype 115
4.5.3 Software Prototype Description 116
4.5.4 Implementation Details 126
4.6 Results and Discussion 130
4.7 Conclusions 131

5 Conclusions 133
5.1 Scientific Conclusions 134
5.1.1 Advancing Data Quality in AI Systems 134
5.1.2 Bridging AI Complexity and User Accessibility 135
5.1.3 Expanding the Frontiers of AI-Powered Decision Support Systems 136

5.1.4 Final Reflections and Future Perspectives 137
5.2 Cross-Chapter Discussions 137
5.2.1 General Conclusions 137
5.2.2 Real-World Applicability 139
5.2.3 Interdisciplinary Insights 140
5.3 Practical Takeaways 141
5.4 Research Perspectives 143
5.4.1 Expanding Application Domains 143
5.4.2 Expanding Real-Time Capabilities 144
5.4.3 Integrating Privacy and Security 145
5.4.4 Automated Enhancements 145
5.4.5 Educational and Training Tools 146
5.4.6 Final Thoughts on Future Research 146

A DSS4EX: Block Descriptions **149**

B DSS4EX: Packages Requirements **153**

C EX-DSS: Keywords **157**

Bibliography **159**

List of publications **171**

List of Figures

1.1 **DSS structure proposed by Turban et al. (2010) [105].** The DSS architecture illustrates the key components that form the foundation of a Decision Support System. The diagram highlights the integration of data, models, knowledge, and user interaction, demonstrating how these elements work together to support effective decision-making. 8

1.2 **High-level visualization of an AI pipeline.** In this thesis, each block represents a distinct design stage carried out by the user through interaction with the DSS. 12

1.3 **Overview of the Decision Support System (DSS) Architecture Proposed in this thesis.** This high-level diagram illustrates the modules that constitute the DSS framework developed in this research. The rectangles represent the two primary modules: the Data Quality Module (DQM), which assesses and enhances data quality, and the Machine Learning (ML) module, which is responsible for managing AI pipeline configuration and deployment. The "Human-in-the-loop" parallelogram signifies the expert's involvement during configuration, where human input not only directs the setup but also provides expert knowledge-driven suggestions to guide decision-making. The ellipses represent key outputs: the deployment phase of the AI pipeline and the resulting outputs from the ML module, which are presented to the user through an interactive visualization interface. This design supports a collaborative process where user feedback refines system outputs, enhancing the DSS's responsiveness and alignment with expert insights. 19

1.4 **Structured overview of the Thesis components.** This visualization illustrates the relationships among the key contributions, chapters, and main modules of the Thesis. The first column summarizes the contributions, while arrows link them to the chapters (second column) where they are discussed in greater detail. The third column highlights the core modules of the DSS, including the Data Quality (DQ) and Machine Learning (ML) modules, along with their connections to the respective chapters. 20

2.1 **Framework architecture overview.** The diagram illustrates the two main concept levels of the framework: the user level (E) and the DSS level (A), along with the key phases involved in the data processing pipeline. The DSS level includes the Intake phase (B), Assessment phase (C), and Analysis phase (D), detailing the subphases and interactions across each stage, from metadata and data upload to the generation of reports and cleaning procedures. 31

2.2 **Example dataset used in the proof-of-concept.** This CSV file contains quarter-hourly offtake and injection measurements for Low-Voltage (LV) grid connections in Belgium. The data gathered from a pilot project conducted in 2016 consists of anonymized readings with over 98% validation. The dataset includes 100 time series of validated measurements from digital meters in a specific Belgian area. Key columns include: `'Type'` (measurement type), `'InstallatieID'` (digital meter ID), `'Afname/Injectie'` (offtake or injection of energy), `'Meter read tijdstip'` (timestamps), `'Eenheid van meetwaarde'` (unit of measurement), `'Meetwaarde'` (measurement value), and `'Status'` (data validity). 37

2.3 **Network Architecture of the DSS.** The DSS follows a client-server paradigm, where multiple client applications communicate with a centralized DSS web server over a network. The web server handles requests for data quality analysis and sends responses to clients. 41

2.4 **Data Processing Flow of the DSS.** This diagram outlines the sequential phases of the DSS: data intake, quality assessment, and interactive analysis. Users provide inputs, receive feedback, and refine the dataset through an iterative process. 42

2.5 **Interaction Between DSS Module and External Database.** This figure depicts how the Data Quality (DQ) Assessment Process interacts with an external relational database (EXT). It shows the data exchange flow and integration between the DSS and the external database, which is used to verify the consistency of the dataset against predefined schema definitions. The diagram is a simplified version of the overall DSS architecture, focusing on the data validation process. 49
2.6 **Example of the Data Cleaning (DC) investigation interface.** This illustrative mock-up presents the conceptual UI of the data quality improvement module developed for the Proof-of-Concept (POC). It includes visual summaries of identified data issues (e.g., missing values, type inconsistencies), proposed historical solutions, and an interactive panel for reviewing metadata and conducting exploratory analyses. The interface demonstrates how the DSS supports the analyst by combining visual feedback and code-based investigation. 54

3.1 **High-Level Overview of the proposed framework. The input to the Decision Support System (DSS) is the data.** The outputs are the Artificial Intelligence (AI) pipeline customized by the user, and the results are obtained by running it with the provided data. These outputs, along with AI suggestions, are presented to the user. The user then decides on the next course of action, which may involve requesting modifications to the pipeline or adjusting hyperparameters for specific blocks. . 69
3.2 **The DSS4EX framework.** The DSS4EX framework represents a Decision Support System (DSS) tailored for systems that aid users in exploring Artificial Intelligence pipelines. It comprises the classical subsystem of a DSS, but the Knowledge Management subsystem and the Model Management subsystem were reinforced with more details specific to the scope. The interaction between the interface and each subsystem is described by the flow of information interchanged. 71
3.3 **Architecture of DSS4EX.** This figure illustrates the client-server architecture of DSS4EX, where the user interacts with the Decision Support System (DSS) through a Graphical User Interface (GUI). The server processes user requests and manages DSS components, while an external containerized service handles the deployment and execution of AI pipelines. The final pipeline configuration is presented to the user based on the processed data and system interactions. 74

3.4 **Processing Flow of DSS4EX.** This diagram outlines the structured interaction between users, the DSS web server, and the external containerized service. The external container is responsible only for deploying the AI pipeline and returning the computed results. However, the DSS4EX software demo extends beyond execution, allowing users to compare multiple pipeline configurations and analyze performance differences. 75

3.5 **Example of DSS4EX interaction in the DR-DNN use case.** The image shows the flow of information for initializing a new project (red), selecting Lags blocks (blue), and training/visualizing results (green). The user creates a project via the interface (A), which is saved in the database (I) through the Data Management subsystem (B). To visualize the pipeline, the interface requests topological configuration from the Model Management subsystem (D) (G). When modifying the Lags block, the system communicates with the Hyperparameter Entity (D.2) and Knowledge Management subsystem (C) for suggestions. Finally, the user triggers the Model Management subsystem (D) to apply the configuration to the time series dataset. 78

3.6 **The Decomposition Residuals Deep Neural Network (DR-DNN) pipeline proposed by Theodorakos et al. (2022) [102].** It is a two-layer pipeline. The first part, the Decomposition Residuals layer, extracts information (trend, seasonalities, and residuals) from the endogenous signal (the variable to be forecasted), which, in this case, corresponds to the electricity consumption. The second part consists of the Non-linear layer, which takes the signals extracted from the Decomposition Residuals layer along with the exogenous signals (independent external variables, like weather, precipitation, etc.) as inputs. It then generates the forecast for electricity consumption as its output. Model explainability is possible via: (1) time series analysis (using classical system identification tools) and (2) ranked input feature importance (using the game-theoretic Shapley values). 80

3.7 **Software Demo Video** This video demonstrates the core functionalities of the software prototype developed using the DSS4EX framework. (Access the video here: `https://youtu.be/1d4e9WgE0g4`) 83

3.8 **New Project Page Interface.** This screenshot displays the initial interface where the user can enter a unique title and description for a new project. The user interacts with this page to create and configure a new pipeline project within the DSS4EX framework. 84
3.9 **Project List and Pipeline Visualization Interface** This screenshot displays the project list page where users can view and select their initialized or completed projects. For each project, the associated pipeline designed by Theodorakos et al. (2022) [102] is visualized, with detailed insights into its two-layer structure: Decomposition Residuals (DR) and Deep Neural Network (DNN). The interface allows users to explore and manage their projects within the DSS4EX framework. 85
3.10 **Pipeline Configuration Interface.** This screenshot illustrates the Configuration Page, where users can read detailed descriptions of each block, receive the designer's suggestions, and configure the block parameters. In the image, it is highlighted the configuration process for the "Lags" block in the DR-DNN pipeline, where users can enable or disable the block and adjust the number of lags based on the designer's recommendations. 87
3.11 **Pipeline Training and External Service Monitoring.** This screenshot shows the interface where the user can monitor the training process of the AI pipeline. The Model Management subsystem triggers the execution on an external service, and the interface provides real-time updates on the training status. The visualization helps users oversee the progress and time efficiency of the pipeline. 88
3.12 **Results Visualization Page.** This figure showcases the results page where the user can visualize the pipeline's outcomes. It includes tabs for time series analysis, feature importance, and forecasting, allowing the user to explore different aspects of the pipeline's performance. 89
3.13 **Overview of the pipeline, results, and SHAP plots of the project Pipeline_without_DR.** Image (a) shows the pipeline after configuration, with grey blocks representing deactivated components related to the DR layer. Image (b) displays the model's predictions alongside expected values for three randomly selected days. 92

3.14 **SHAP Plot for Pipeline_without_DR.** This plot highlights the ranking of the most significant features using feature importance. In this scenario, the SHAP plot indicates that features related to "Load(kW)" dominate as significant, with "Temperature (C)" being the sole exogenous variable of importance. An embedded expert comment interprets this limited feature set, noting a primary focus on load and one exogenous signal. The comment further suggests that incorporating additional external signals could improve the model's robustness. 93

3.15 **Overview of the pipeline, results and SHAP plots of the project Pipeline_DR-DNN.** Image (a) presents the pipeline after configuration. The grey block, specifically the "λ parameters from tests", is the deactivated block. Image (b) plots the predictions found by the model with their expected values for three randomly selected days. 94

3.16 **SHAP Plot for Pipeline_DR-DNN.** This plot This SHAP plot reveals the most influential feature and a more complex feature interaction, where a range of signals, including multiple types of signal decompositions and additional exogenous variables, are used. The tool provides an expert comment here as well, which guides users by explaining that the decomposition residual block allows the model to capture more nuanced influences from diverse features, thus enhancing prediction accuracy. "Load(kW)_trend_23h" emerges as the most influential feature. 95

3.17 **Details of the Comparison Page: Top View** The image displays the top of the web page. It includes the selection of the time slots and plot of the predictions obtained by running Pipeline_without_DR and Pipeline_DR-DNN with the expected values. 96

3.18 **Details of the Comparison Page: Bottom View** The image shows the bottom of the page. It comprises the residual plot, so the difference between the expected values and the predictions calculated by the pipelines, Pipeline_without_DR and Pipeline_DR-DNN. Some error metrics (Mean Squared Error, Mean Absolute Error, etc.) are also calculated to provide more insights. The residuals plot shows that the Pipeline_without_DR over-estimates the forecasted values. 97

3.19 **Architecture of DSS4EX** This figure illustrates the technological interactions within DSS4EX. The user interacts with the Decision Support System (DSS) through a Graphical User Interface (GUI) developed in Dash. The GUI retrieves information processed by the server, which hosts DSS components using the web-server Flask. The configured AI pipeline is deployed in a Docker container managed by Apache Airflow. In the end, the user is shown a personalized pipeline resulting from the requests made and based on the retrieved information from the Apache Airflow service. 98

4.1 **EX-DSS architecture framework.** This illustrates the proposed EX-DSS framework. (A) User-level input and outputs. (B) Blocks composing the EX-DSS: (C) Block to assess the quality of the user's input and (D) Block to aid the users in designing, configuring, and deploying the pipeline. (E) User Interface that allows the user to dialogue with the EX-DSS. The arrows represent the flow of information inside the systems. . . 109

4.2 **Processing Flow of the EX-DSS Framework.** The diagram represents the structured interaction between users, the DSS web server, and the AI pipeline execution. Users iteratively refine and explore different AI configurations before finalizing results. 113

4.3 **EX-DSS Configuration Page - List of Available Blocks.** This screenshot showcases the blocks developed within the EX-DSS software prototype. Users can choose from these blocks to design their data processing and forecasting pipeline. 121

4.4 **EX-DSS Configuration Page - Block Suggestions.** The system provides users with tailored suggestions for each block, which include a description, expert insights based on the project's keywords, and a communication interface with the Knowledge Management Subsystem. These suggestions guide users in selecting appropriate methods and techniques for their dataset. 122

4.5 **EX-DSS Configuration Page - Block Configuration Area.** This section allows users to configure parameters for each selected block. In the shown example, the user configures the "Forecast" block by selecting the forecasting method, "Global XGBoost," and setting the prediction horizon to one day. 123

4.6 **EX-DSS Configuration Page - Design Area.** The design area allows users to organize and connect blocks to define the data processing and forecasting pipeline. In this example, the blocks for dataset selection, appliance operating mode, forecasting, scheduling, and evaluation are connected, while "Data Cleaning" and "Preprocessing" are omitted as the dataset is already preprocessed. 124
4.7 **Airflow Server Runtime Screenshot.** This screenshot displays the Airflow service used by the EX-DSS software prototype to execute the pipeline. It shows the runtime details of the training process, which took 9 minutes and 24 seconds to complete. 125

List of Tables

2.1 **Results of the Interoperability Check.** ThisTable presents the outcome of the interoperability check performed on the data and metadata during the EU guidelines assessment process. The check, conducted using the cChardet library, ensures that no encoding issues are present in the files. In this case, the system found no encoding issues, confirming the integrity of the data. . 46

2.2 **Metadata Quality Indicators.** This table shows the results of the text quality analysis applied to the metadata description during the EU guidelines assessment process. Indicators such as Spelling Errors, Lexical Diversity, Ungrammatical Sentences, and Average Sentence Length were calculated. The analysis provides insights into the clarity and reusability of the metadata, with a low lexical diversity indicating potential redundancy in word usage. 47

2.3 **Consistency Check Results.** This table presents the results of comparing the dataset columns with the expected types defined in the external relational database. The system checks the compatibility between each dataset column and the corresponding attribute in the external database. The Table lists column names, the attribute types found in the external database, the types identified in the dataset, and the result of the match (Success or Fail). 49

2.4 **Missing Values and Rows.** This table displays the results of the system's analysis of missing values and rows in the dataset. The first column shows the types of missing data checked, while the second column indicates the number of occurrences of each type. The analysis includes both null values and missing rows based on the provided timestamps and metadata. 50

2.5 **Data Cleaning Indicator Calculation.** This table summarizes the results used to compute the Data Cleaning (DC) indicator. It lists the types of cleaning issues identified, the number of actual problems found, and the number of problems checked by the DSS. The DC indicator is computed by comparing the number of checked problems with the actual problems that require further user intervention. 51

3.1 **Snapshot of the dataset used in the software demo.** It is a public time series dataset called "Day-Ahead Electricity Demand Forecasting: Post-COVID Paradigm" [38]. The time series dataset starts on the 18th of March 2018 at 8 a.m. and ends on the 16th of February 2021 at 11 p.m., and the signals are collected hourly. It comprises the endogenous signal "Load (kW)," the city power load, and the exogenous signals "Pressure kpa," "Cloud Cover," "Humidity," "Temperature," "Wind Direction," and "Wind Speed." 82

3.2 **Comparison of Configuration Parameters.** This table compares the parameter values of two projects, Pipeline_without_DR and Pipeline_DR_DNN, highlighting the differences in configuration, such as the activation of specific blocks and their parameter settings. 87

3.3 **Hyperparameter Table for Pipeline Configuration** This table presents the hyperparameter values for the selected pipeline configuration, detailing the adjustable parameters for each block. 90

4.1 **Snapshot of the Smart Plugs Preprocessed Dataset.** The dataset consists of power consumption data from 169 smart plugs, each monitoring various electrical appliances. Power values are recorded in milliwatts (mW) at fifteen-minute intervals, spanning from November 18th, 2021, to March 31st, 2023. The table displays a subset of the data, including time series for 146 printers, 16 copiers, 4 TVs, and 3 fax machines. 117

4.2 **Schedule Overview.** The table provides an example of the schedule for plugs 0 and 168. Both plugs are off at night and turn on in the morning, with plug 0 turning on earlier. Plug 0 turns off around lunch while plug 168 turns off later in the afternoon. The schedule can be summarized with the turn-on and turn-off times: Plug 0: {2023-01-13 05:15:00: "Turn on"; 2023-01-13 12:30:00: "Turn off"}, Plug 168: {2023-01-13 09:30:00: "Turn on"; 2023-01-13 16:45:00: "Turn off"} 127

4.3 **Forecasting Evaluation Metrics.** This table summarizes the performance of the "Global XGBoost" forecasting method in the pipeline, including key metrics such as the number of violations, missed chances, energy saved, and energy efficiency. 127

A.1 **Block Descriptions in the DR-DNN Pipeline.** This table lists and describes each block used in the Decomposition Residual - Deep Neural Networks (DR-DNN) pipeline, which is presented in Chapter 3. The descriptions are shown to the user during the pipeline configuration phase to provide clarity on each block's function and purpose. 149

B.1 **Python Packages Used in the DSS4EX Framework.** This table lists the Python packages utilized in the development of the DSS4EX prototype, including their respective versions and descriptions. These packages were selected to ensure compatibility, stability, and reproducibility, which are essential for the framework's functionality, as detailed in Chapter 3. The table also includes links to the package documentation for reference. 153

Chapter 1

Introduction

1.1 Overview

This dissertation addresses the improvement of interactions between humans and artificial intelligence (AI) systems. It introduces frameworks designed to build trustworthy software systems accessible to both experts and non-experts who aim to design or understand complex AI models. These frameworks are tested using practical examples from time series scenarios, demonstrating their functionalities. For clarity, this thesis defines "framework" as an architectural framework, i.e., the structural design and organizational blueprint of a software system. This set of guidelines provides principles and practices for creating and using a system's architecture description.

The first part of the Thesis introduces a framework for a Data Quality Module (DQM) for a Decision Support System (DSS). The primary objectives are to highlight the importance of Data Quality (DQ) in generating high-quality outcomes and to expedite the data quality assessment process. The proposed DQM utilizes various guidelines, such as European (EU) standards, to compute two scores: one indicating the overall level of DQ, answering "How good is the input dataset?" and another assessing the ease of Data Cleaning (DC), answering "How easy is it to clean the input dataset?" These scores provide immediate insight into the dataset's impact and the effort required to prepare it for subsequent tasks. This DQM is demonstrated through a proof-of-concept using a public time series dataset, which includes quarter-hourly consumption profiles of residential electricity customers in Belgium for the year 2016.

The second part of the Thesis presents a framework for a DSS for exploring

(DSS4EX) AI pipelines. DSS4EX is designed to develop educational software tools for users seeking to understand complex AI pipelines. These tools enable users to explore AI pipelines created by experts in specific fields, configure the pipeline, visualize and download results, and compare parameters. The software tool guides users by providing descriptions and suggestions. A practical example of DSS4EX is demonstrated through a software prototype showcasing the Decomposition Residuals Deep Neural Network (DR-DNN) pipeline, developed by Theodorakos et al. (2022) [102], and tested with an open dataset for energy consumption forecasting.

The final part integrates the previously proposed frameworks into the EX-DSS, which stands for Explorative DSS. This comprehensive framework consists of two steps: the DQ module and the ML module. The DQ module is based on the first part of this thesis, while the ML module extends the capabilities of DSS4EX. The ML module enhances DSS4EX by offering greater support to end-users and flexibility during the configuration phase. EX-DSS aims to assist users in designing and training complex AI pipelines. The work's applicability is demonstrated through a real-world scenario: a software prototype featuring methods from a novel pipeline for energy savings in office buildings, focusing on plug mode detection, forecasting, and scheduling, proposed by Botman et al. (2024) [8].

In this introductory chapter, we explore the importance of data quality and the role of educational AI tools in fostering trustworthiness around AI systems. We also discuss the scientific significance of this research.

1.2 Context

This section aims to establish a clear understanding of Decision Support Systems (DSS) by outlining the theoretical foundations that support their design and development. By providing this foundation, the term DSS is contextualized within the scope of this research, ensuring a shared understanding from which to differentiate DSS from related concepts. The context presented in this section offers an overview of the key concepts essential for understanding the subsequent chapters, setting a solid groundwork for exploring the complexities and applications of DSS throughout the Thesis.

1.2.1 Introduction to Decision Support System

DSS belongs to the broader family of Information Systems (IS), which are sets of programming tools designed to process and manage large volumes of data, enabling users to visualize and act upon the information efficiently [72]. These systems intersect the fields of computer science and business, making them pivotal in data-driven decision-making environments [35]. Within IS, notable systems include Expert Systems (ES), Management Information Systems (MIS), and Decision Support Systems (DSS) [1].

ES are specialized tools that simulate expert knowledge to solve specific problems [1]. These systems are heavily problem-driven and often rely on Artificial Intelligence (AI) to replicate human decision-making processes [1].On the other hand, MIS focus on organizing and presenting historical, current, and forecasted information to facilitate the planning, control, and operational activities within organizations, delivering reports tailored to managerial needs [4].

In contrast, DSS offer tailored support to decision-makers by providing relevant data, analytical tools, and models to assist in solving specific problems [76, 4]. Different from ES and MIS, DSS are more flexible because not only can they incorporate characteristics specific to ES and MIS, they also support a broader range of decisions by analyzing data and providing comprehensive insights. As a matter of fact, a DSS allows for the incorporation of characteristics of ES and MIS like, for example, leveraging rule-based expertise within a domain, generally associated with ES, and delivering routine operational reports, normally attributed to MIS. Through its interactive interface, a DSS allows users to explore insights, interpret results, and make more informed decisions. Particularly, DSS is designed to aid unstructured or semi-structured decisions, as all the IS [104, 113, 1]. Unstructured decisions lack predefined solutions and rely on human intuition and judgment [104, 113]. Semi-structured decisions, meanwhile, involve a combination of human input and mathematical models, where DSS enhances decision-making by integrating these two elements [104, 113].

With the advent of more sophisticated technologies, Intelligent Decision Support Systems (IDSS) have emerged, integrating AI techniques like machine learning and natural language processing [71, 61]. These systems enhance traditional DSS capabilities by learning from data patterns, providing predictive analysis, and offering more accurate and real-time support in decision-making processes [71]. AI-driven DSS are invaluable in dynamic, high-stake fields such as healthcare, finance, and logistics, where decisions must adapt quickly to evolving real-time data and uncertainties [71, 33]. By augmenting human judgment with advanced AI-driven tools, IDSS accelerate the generation of actionable insights

and improve decision outcomes in both structured and unstructured scenarios [71]. For the purpose of this investigation, no distinction between the terms IDSS and DSS will be made because the current need for more human-in-the-loop systems signifies the use of AI to enhance DSS.

Historical Evolution of DSS and DSS Applications

The origins of Decision Support Systems (DSS) can be traced back to the late 1960s and early 1970s when computer technology enabled the development of systems to improve managers' decision-making processes [73, 74]. Early DSS focused on model-driven approaches, providing users with mathematical and statistical models to assist in quantitative analysis [73]. One of the key turning points in the evolution of DSS was the introduction of spreadsheet software, such as VisiCalc, in the early 1980s, which democratized data analysis by making these tools accessible to non-technical users [73].

The 1980s and 1990s saw significant developments in DSS, particularly with the rise of data-driven systems that leveraged growing database and data warehousing technologies. These systems allow users to analyze historical data, generate reports, and use vast data resources to inform their decisions [73, 74]. During this period, the emergence of knowledge-driven DSS incorporated artificial intelligence (AI) and expert systems, making it possible for decision-makers to receive insights and recommendations based on pre-configured knowledge bases [73, 74]. The introduction of communications-driven DSS added a collaborative element, allowing teams to work together on decision-making processes using groupware and conferencing tools [74].

A variety of reference disciplines and technological advancements have influenced the evolution of DSS. Eom [34] outlines that reference disciplines such as computer science, management science, organizational theory, cognitive psychology, and artificial intelligence have significantly shaped DSS. The interdisciplinary nature of DSS reflects its complex integration of technology, decision theory, and human-computer interaction. From a technical perspective, DSS build upon advances in databases, information systems, and software engineering, while management science provides foundational principles in decision theory, optimization, and systems thinking. Cognitive psychology has contributed insights into human decision-making behavior, allowing DSS to align better with how users process information and make choices. Incorporating AI has further opened up possibilities for Intelligent Decision Support Systems (IDSS), which employ machine learning and predictive modeling techniques to offer sophisticated decision aids [61].

Moreover, Shim et al. (2002) [88] highlight the shift in DSS technology from supporting individual decision-makers to enabling workgroups and even inter-organizational systems driven by web-based capabilities. This evolution is marked by advancements in database management, modeling capabilities, and user interfaces, expanding DSS applications and making them accessible to broader audiences.

As technology continues to evolve, so do DSS applications. The 1995–2001 period saw DSS being applied to various sectors, including healthcare, finance, and manufacturing [35]. By 2012, DSS development trends pointed to broader applications, particularly in education, government, and supply chain management [52]. Clinical Decision Support Systems (CDSS) have emerged as crucial tools in healthcare, helping clinicians diagnose and treat patients more effectively by offering real-time data analysis and decision recommendations based on clinical guidelines and patient history [99].

DSS continues to play a pivotal role in sectors such as finance, logistics, and strategic management, helping organizations optimize operations, make informed investment decisions, and manage risks more effectively [4]. With the integration of AI, DSS applications are extending into real-time decision-making in dynamic environments, including smart cities, autonomous systems, and advanced manufacturing [61, 52]. As DSS continue to evolve, its applications in decision-making contexts will further expand, driven by advancements in data analytics, AI, and cloud computing.

Types of Decision Support Systems

DSS are categorized into various types based on their functionalities and the nature of decision support they provide. The major types include Data-Driven, Model-Driven, Knowledge-Driven, Communications-Driven, and Document-Driven DSS [39].

- **Data-Driven DSS** focuses on accessing and analyzing large datasets to help users make informed decisions based on historical data, trends, and real-time analytics [39]. These systems utilize tools such as data warehouses, online analytical processing (OLAP), and business intelligence platforms to provide relevant data for decision-making [86]. In this context, the focus is on large-scale data collection and aggregation, supporting decision-making by delivering processed data in a comprehensible form [39].
- **Model-Driven DSS** center around the use of mathematical and statistical models. These models support decisions by providing analytical

tools for scenarios that involve optimization, simulation, financial modeling, or other quantitative aspects of decision-making [113, 39]. Unlike data-driven systems, the emphasis here is on manipulating models and algorithms to simulate outcomes and explore decision scenarios [76].

- **Knowledge-Driven DSS**, sometimes called "intelligent" DSS, are based on artificial intelligence (AI), machine learning, and expert systems [71]. These systems draw on domain-specific knowledge and rules to generate tailored user recommendations. They emulate human expertise in specific problem domains, offering advice based on predefined knowledge bases and learning from new data over time to improve the quality of decision support [105].

- **Communication-Driven DSS** are designed to facilitate collaborative decision-making. These systems provide tools for group discussions, online collaboration, and information sharing, often through groupware, video conferencing, and collaborative platforms [39]. The main goal is to support team-based problem-solving and improve the coordination of decision-making processes across teams or departments [39].

- **Document-Driven DSS** manage and retrieve unstructured information in various formats (text, images, audio, video) to support decision-making processes [39]. By accessing, organizing, and analyzing documents and other content, these systems enable better understanding and facilitate decisions based on non-quantitative data sources [39].

In this thesis, the framework proposed is designed to be versatile and adaptable without focusing on a specific type of Decision Support System (DSS). This flexibility is intentional, as the framework aims to be applicable across various DSS types, whether Data-Driven, Model-Driven, Knowledge-Driven, or Communications-Driven. However, the nature of the applications and case studies explored in this work align more closely with Model-Driven and Knowledge-Driven DSS. These types are particularly relevant given their reliance on complex modeling, simulation, and expert system-based recommendations.

Components of Decision Support Systems

A well-functioning DSS comprises several key components that work together to support decision-making:

- **Data Management Component**: This component facilitates the collection, storage, retrieval, and management of relevant data from

a variety of sources, such as databases, data warehouses, and online sources [39]. It ensures that users have access to accurate, timely, and comprehensive information required for informed decision-making [76].

- **Model Management Component**: This component consists of analytical models, algorithms, and simulation tools that allow users to perform scenario analysis, forecasting, and trend prediction [113]. It enables the decision-maker to analyze different alternatives and evaluate the potential outcomes of various decision scenarios, enhancing the understanding of complex issues and aiding strategic planning [76].
- **Knowledge Management Component**: Especially relevant in knowledge-driven DSS, this component integrates AI and expert systems to replicate human decision-making processes and provide domain-specific advice and recommendations [113]. It acts as a repository of structured rules and knowledge that can be applied to support complex decisions, particularly where expertise is critical [71].
- **User Interface Component**: The User Interface (UI) is the interaction point between the system and its users. A user-friendly interface is crucial for ensuring accessibility to both technical and non-technical users [76, 113]. It allows for easy data input, model selection, and visualization of results, enhancing the DSS's usability and effectiveness [76].

Figure 1.1 provides a visual representation of the DSS components discussed, illustrating how data management, model management, knowledge management, and the user interface interconnect to form a cohesive system [105]. This visual aid helps clarify the interactions between the components, demonstrating how they work together to support decision-making processes [105]. Together, these components form a comprehensive system that supports decision-making through data analysis, model-based simulations, and expert recommendations [76, 113]. Integrating data, models, and a user interface allow a DSS to effectively support structured and unstructured decision-making scenarios [39]. The integration of these elements is believed to constitute a comprehensive interactive system and be conclusively necessary when designing a DSS.

Decision-Making Model

Simon [13, 63] proposed a decision-making model during the 1960s which has continued to be widely used to this day. The author describes decision-making as a process involving intelligence, design, and choice phases, all of which have been influential in understanding decision-making dynamics [13]. Simon [13] emphasized the concept of "bounded rationality," acknowledging the cognitive

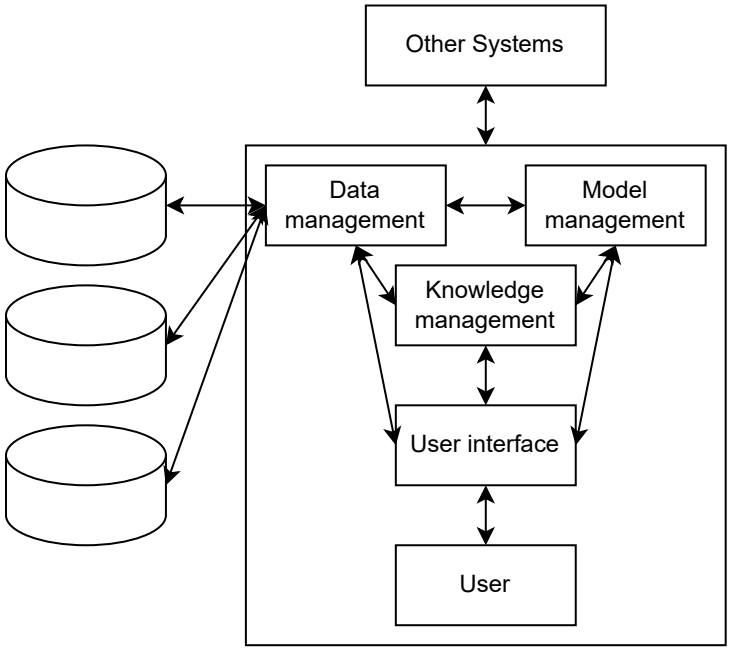


Figure 1.1. **DSS structure proposed by Turban et al. (2010) [105].** The DSS architecture illustrates the key components that form the foundation of a Decision Support System. The diagram highlights the integration of data, models, knowledge, and user interaction, demonstrating how these elements work together to support effective decision-making.

limitations of human decision-makers in processing information and making optimal decisions. This emphasizes the need for external support that can help users overcome these limitations and so, be better able to assess more critically a situation, analyse the different perspectives or angles that there may be, in order to make informed decisions. While Simon's model makes relevant contributions on the cognitive aspect of decision making, this thesis focuses on a more practical application, which means that the proposed framework is designed from a software-oriented perspective, focusing on the technical and architectural aspects of a DSS development. Therefore, the intention is to provide a practical, adaptable framework that can integrate with different types of DSS, emphasizing technological implementation over the theoretical decision-making process itself.

1.2.2 DSS and Importance of Data Quality

Data quality (DQ) is a critical factor in DSS and AI, directly influencing the performance and accuracy of these technologies. DQ refers to the state of the data's qualitative characteristics for an intended use and is defined by several key dimensions, including accuracy, completeness, consistency, and timeliness. **Accuracy** ensures that data is error-free and represents real-world conditions accurately [89, 30, 78]. **Completeness** involves having all necessary data points and avoiding missing values that could alter analysis or decision-making [89, 30, 78]. **Consistency** means that data is uniform across different sources and datasets, reducing discrepancies that may lead to unreliable outcomes [89, 30, 78]. Finally, **timeliness** emphasizes the importance of up-to-date data, ensuring that decisions are based on the most recent and relevant information available [89, 30, 78].

In the context of AI and DSS projects, DQ plays a pivotal role in achieving successful outcomes. High-quality data enhances the performance and reliability of AI models by providing a solid foundation for training and evaluation. When AI models are trained on accurate, complete, and consistent data, they are more likely to generate trustworthy predictions and insights. In DSS, high-quality data helps facilitate informed, data-driven decisions, thereby improving the overall decision-making process within organizations [42]. Conversely, poor DQ can lead to flawed models and incorrect decisions, undermining the effectiveness of AI and DSS. Studies have shown that incorporating DQ information can significantly impact decision makers confidence and the outcome of their decisions [41, 18].

Ensuring DQ in AI and DSS projects poses several challenges despite its importance. DQ issues, such as errors, missing values, and inconsistencies are common in large and complex datasets. Data cleaning and preprocessing are necessary steps to address these issues, but they can be time-consuming and require specialized tools and methodologies. Furthermore, maintaining DQ becomes increasingly difficult as the volume, velocity, and variety of data grow. The need for efficient DQ measurement and monitoring tools has become apparent to ensure that AI and DSS can operate effectively [30]. In AI, especially deep learning, the DQ used for model training is crucial since poor data collection and preparation can lead to suboptimal model performance [109]. Therefore, developing frameworks and methodologies to assess and monitor DQ is essential for building trustworthy AI systems [85]. Addressing these DQ challenges is critical to harness the full potential of AI and DSS to support complex decision-making processes.

1.2.3 DSS and Trustworthiness of Intelligent Systems

Trustworthiness in AI systems is an essential aspect of their design and adoption. It encompasses a user's reliance on the system's ability to function accurately, ethically, and fairly, particularly in high-stake contexts like healthcare, finance and education [91, 53]. The success of AI technologies greatly depends on their perceived trustworthiness, which affects how willing users are to depend on the AI's outputs and recommendations. In this thesis, the problem of trustworthiness is addressed by developing a software framework that is configurable in every aspect, offering modularity to allow developers to customize the system and enhance its accessibility for end users. By taking this approach, transparency, straightforwardness and a user-centered design are prioritized, which is believed to increase trustworthiness in the system.

The trustworthiness of AI is built upon multiple key factors, including transparency, explainability, fairness, and robustness [53]. Transparent AI provides insight into how the system processes data and makes decisions, while explainable AI seeks to present a logical narrative of the decision-making process [51]. Fairness is also critical; it ensures that AI systems do not perpetuate biases and that decisions are equitable across different user groups [60]. Additionally, robustness and privacy are fundamental. In fact, AI systems must be able to handle diverse inputs, protect sensitive user data, and remain resilient against adversarial attacks [83]. Our framework, as demonstrated through prototypes, emphasizes clarity throughout the analysis process, providing access to parameters and comprehensive explanations of how results were computed. In this thesis, the privacy aspect is not examined but it is included in future works, and it is crucial when the prototype will be designed for production.

Current challenges in achieving trustworthy AI systems revolve around the complexity and lack of transparency in models like deep learning, data biases that can lead to unfair outcomes, and robustness to unexpected situations [60]. Solutions such as Explainable AI (XAI) and Controllable AI aim to enhance user comprehension and offer more control over AI behavior [51]. Moreover, techniques for detecting and addressing biases are being actively developed to ensure fairer outcomes. Robust model training and privacy-preserving techniques are also vital to build resilience and secure user trust [83]. In this thesis, our proposed framework's modularity allows the software to be adjusted to address these challenges effectively, making the system more accessible and trustworthy.

Another aspect that helps in having trustworthiness AI-systems is the development of user-friendly systems [103]. User-friendliness makes AI technologies accessible to a wider audience, including non-experts. Educational Data Mining (EDM) and Learning Analytics (LA) have emphasized creating

intuitive tools that educators and learners can use without requiring advanced technical knowledge [91, 26]. In the work presented, user-friendliness is achieved through intuitive interfaces that guide users through configuration and provide expert suggestions. The prototype presented in Chapter 2 serves as an educational tool by leading users through the setup phase, offering explanations and insights that help them understand the system's behavior and the analysis process [79].

User-friendly AI tools are crucial for driving adoption and fostering innovation. When AI is made approachable and usable for various stakeholders, such as educators, domain experts, and policymakers, it encourages the spread of AI technologies and opens up opportunities for creative applications and problem-solving across sectors [91, 80]. Tools like Microsoft Excel[1], WEKA[2], RapidMiner[3], and Knime[4], which are widely used in educational contexts, make complex data analyses accessible through visualizations and interactive data exploration. In our prototypes, the user-centered design of the software facilitates deeper understanding and allows users to actively participate in data-driven decision-making.

In addition, educational tools play a pivotal role in enhancing trust in AI by increasing user comprehension and confidence [80]. By offering intuitive, guided configurations and expert suggestions, our framework not only supports educational objectives but also contributes to building trust in AI systems. The modularity and adaptability of the framework ensure that users can customize the system to meet their specific needs, whether for educational purposes or other domains.

In summary, our Thesis targets the problem of AI trustworthiness through a configurable software framework that prioritizes user access and understanding. With its educational components, clear explanations, and emphasis on transparency, our work aims to contribute to making AI systems both trustworthy and user-friendly, supporting the broader adoption and effective use of AI technologies across various domains.

1.2.4 DSS and Artificial Intelligence Pipelines

AI pipelines are essential frameworks that automate the entire machine learning (ML) workflow, facilitating the processing of data through various stages. These pipelines ensure that models are trained, evaluated, and deployed efficiently

[1] `https://www.office.com/`
[2] `https://www.weka.io/`
[3] `https://altair.com/altair-rapidminer`
[4] `https://www.knime.com/`

and reproducibly. Typically, an AI pipeline begins with data collection and preprocessing, where raw data is cleaned and transformed into a suitable format for analysis. This is followed by model training, during which machine learning algorithms learn patterns from the data. Once trained, models undergo evaluation to ensure they meet performance criteria before being deployed into production systems [94, 97].

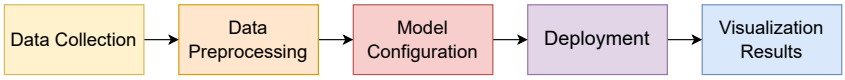


Figure 1.2. **High-level visualization of an AI pipeline.** In this thesis, each block represents a distinct design stage carried out by the user through interaction with the DSS.

However, in this thesis, it is important to emphasize a key distinction: unlike the cited works, the deployment phase here refers to model training that is done automatically by the DSS before the evaluation phase. The term "deployment" is used to signify that this step occurs externally from the DSS, as it would in a production system. Figure 1.2 illustrates the components of the AI pipeline as considered in this thesis. AI pipelines play a critical role in maintaining the continuous development of models, especially in complex environments that demand high adaptability and accuracy [102, 8, 79]. These pipelines have seen widespread adoption across various domains. In healthcare, AI pipelines enhance clinical decision-making by automating diagnostics and risk prediction, often outperforming traditional methods in terms of speed and accuracy [31]. In industrial settings, they are vital for tasks such as predictive maintenance and energy optimization, where real-time data processing is crucial for operational efficiency [2]. Moreover, AI pipelines ensure scalability, enabling AI models to process large datasets while maintaining reproducibility, which is vital for ongoing research and development [100]. The importance of AI pipelines extends beyond automation. They provide a structured approach to managing data flow, model selection, and hyperparameter tuning, thereby optimizing the performance of AI solutions [97]. By automating repetitive tasks, pipelines also free up human resources, allowing for more strategic decision-making [100]. Furthermore, pipelines ensure that models are regularly updated with new data, which is particularly valuable in domains that require dynamic learning systems, such as cybersecurity and anomaly detection [2].

However, designing and implementing AI pipelines comes with significant challenges. One of the primary hurdles is the selection of appropriate models and algorithms. Ensuring the correct preprocessing of data and effective feature extraction is critical for model performance [94, 69]. Another challenge lies in hyperparameter tuning, which must be optimized to maximize the accuracy and efficiency of the model [69, 102]. Additionally, managing the flow of large volumes

of data can be complex, especially when integrating models into production systems that require continuous monitoring and updates [102, 79, 97]. The ability to handle these challenges effectively determines the success of the AI pipeline in delivering accurate, scalable, and reproducible results. In this thesis, these challenges are addressed using DSS [79].

Given their ability to automate complex workflows while maintaining adaptability, AI pipelines are essential for any organization leveraging AI to improve decision-making processes [100, 24]. This importance becomes even more pronounced in specialized applications such as time series forecasting, where managing sequential data adds an additional layer of complexity [102, 8]. In this thesis, the focus is on time series data and time series forecasting pipelines.

Time series data is distinct in its sequential nature, where each observation is indexed in time, making the temporal structure a crucial factor in analysis. Applications of time series data span across various domains, such as forecasting energy consumption, predicting financial market trends, and managing operational demands in real-time systems [57, 45].

Given the complexity of time series data, numerous techniques have been developed to forecast future trends and patterns. While traditional approaches like autoregressive models and ARIMA remain prevalent, modern hybrid models like the Decomposition-Residuals Deep Neural Network (DR-DNN) have demonstrated significant improvements in capturing both long-term and short-term dynamics [62, 102].

However, the focus of this thesis is not on the detailed implementation of these models, but rather on how AI pipelines can simplify and automate the generation of such forecasting techniques. By leveraging DSS, this work aims to streamline the creation and deployment of time series forecasting models, making these advanced techniques accessible without requiring in-depth technical expertise.

1.2.5 DSS and Current Technologies

DSS continue to be pivotal in enabling effective decision-making across diverse domains, such as energy, healthcare, manufacturing, and finance. Advances in Artificial Intelligence (AI), including machine learning (ML), data analytics, and automation frameworks, have redefined the design and functionality of DSS. Modern DSS are no longer limited to traditional structured decisions but are increasingly capable of tackling semi-structured and unstructured scenarios by integrating AI techniques [98]. This transformation is driven by the emergence of

technologies such as Generative AI and Automated Machine Learning (AutoML), which enhance the accessibility, usability, and performance of DSS [17, 75].

The significance of DSS lies in their ability to bridge human expertise with advanced AI capabilities, providing insights through data-driven analyses while maintaining human-in-the-loop interactions [98]. Current technologies enable DSS to handle vast amounts of heterogeneous data, optimize workflows, and ensure real-time decision-making while fostering transparency and user trust. In sectors such as Industry 4.0 and Industry 5.0, DSS augmented with AI improve operational efficiency, predictive analytics, and collaboration between humans and machines [75, 111].

DSS and Generative Artificial Intelligence

Generative Artificial Intelligence (GenAI), with its ability to produce novel and context-specific content, has become a transformative component of DSS. By leveraging models such as Generative Adversarial Networks (GANs) and large language models (LLMs), DSS can generate synthetic data, simulate decision scenarios, and automate insights generation [17, 64]. In financial forecasting, for instance, GenAI enables the simulation of market conditions and risk scenarios, significantly enhancing decision accuracy and robustness [17]. Similarly, in healthcare, generative models facilitate clinical decision support by synthesizing patient data and generating actionable treatment recommendations [10].

GenAI also fosters natural language interaction within DSS, allowing users to query systems in an intuitive, human-like manner. This capability enhances system usability for non-expert users and promotes trust by generating interpretable explanations of system outputs [64, 12]. In education and research, GenAI assists decision-makers by providing tailored learning materials, automating content creation, and enabling scenario-based learning [12].

In this thesis, Chapter 4 introduces the integration of a GenAI model to generate text-based responses within the DSS. This feature supports decision-makers by answering general questions in a natural language format, improving system accessibility and usability for end-users.

DSS and AutoML

Automated Machine Learning (AutoML) represents a significant advancement for DSS, particularly in automating the end-to-end machine learning pipeline, including data preprocessing, model selection, and hyperparameter optimization [75, 112]. AutoML enables DSS to democratize AI capabilities by reducing

the need for specialized technical expertise, thereby making machine learning accessible to domain experts [56]. In dynamic and complex environments, such as smart manufacturing and cybersecurity, AutoML-powered DSS facilitate rapid decision-making by automating the creation and deployment of optimized models [75, 112].

One of the major benefits of integrating AutoML into DSS is the scalability and adaptability it offers. For example, AutoML enhances data pipelines by automating the flow of data between different stages of machine learning workflows, resulting in efficient model development [112]. Additionally, AutoML can optimize decision support in Industry 4.0, where real-time monitoring, anomaly detection, and predictive maintenance are crucial [75].

However, despite its advantages, AutoML still faces challenges related to transparency, interpretability, and user control. Current systems often prioritize predictive performance over explainability, which can undermine user trust [56]. Furthermore, AutoML frameworks may lack flexibility in incorporating domain knowledge and iterative feedback, which are critical for decision-making processes [56, 75].

In this thesis, the use of AutoML was intentionally excluded from the framework design to focus on demonstrating the structure and usability of DSS. The primary objective was to help users understand and interact with AI models rather than automating every aspect of the pipeline. Nevertheless, integrating AutoML into future iterations of the DSS could be valuable for automating processes and optimizing model selection, particularly when the scope involves full end-to-end automation of decision workflows.

1.3 Research Objectives

Main Goal
To design, implement and assess Frameworks for Designing Decision Support Systems to Generate Artificial Intelligence Pipelines for energy time series applications.

Supporting Objectives
The different chapters in this thesis contribute to the achievement of the general aim by:

- Developing a user-friendly framework for DSS that aids in the design and deployment of AI pipelines.

- Introducing an innovative DQM framework that assesses and improves data quality, enhancing AI model performance.
- Integrating the DQM and DSS4EX frameworks into a comprehensive EX-DSS system that supports the entire AI pipeline lifecycle.
- Demonstrating the effectiveness of the DQM, DSS4EX, and EX-DSS in real-world scenarios by testing them in smart plug forecasting and power load energy forecasting use cases.

1.4 Methodology

The research methodology of this thesis involves a combination of theoretical and practical approaches to systematically design, implement, and evaluate the proposed frameworks for Decision Support Systems (DSS). The methodology is divided into six main stages:

1. **Literature Review**: A comprehensive review of relevant literature was conducted to understand existing frameworks and methodologies in DSS, DQ, and AI pipeline design. The literature review focuses on current challenges, limitations, and best practices in DSS and AI-driven decision-making to identify areas for improvement.
2. **Framework Design**: Based on insights from the literature, two foundational modules were designed: a Data Quality Module (DQM) framework and a DSS framework for AI exploration (DSS4EX). Each module was structured to address specific challenges within DSS, focusing on modularity, user accessibility, and the promotion of trustworthy AI by enhancing transparency and user engagement. The EX-DSS framework was subsequently developed by integrating these modules into a comprehensive architecture.
3. **Prototype Development**: Software prototypes for each module and the EX-DSS framework were developed to validate the functionality of the proposed designs. The prototypes were created using established software development practices, focusing on modularity, ease of use, and educational value.
4. **Use Case Implementation**: To demonstrate real-world applicability, the frameworks were implemented in a use case for energy consumption forecasting and smart plug load management in office buildings. The DQM was used to assess and pre-process the datasets, and DSS4EX provided an

interface for configuring and deploying AI pipelines for predictive analysis. The use case also highlights the importance of DQ and showcases the framework's user-guided approach.

5. **Prototype Evaluation**: Each prototype was evaluated through a series of prelimary informal validation experiments to assess their effectiveness in supporting decision-making, improving data quality, and enhancing user experience. This involved comparing predictions from the AI pipelines designed through DSS4EX with those generated by traditional methods to measure the impact on accuracy, efficiency, and user satisfaction. Metrics such as data pre-processing time, ease of use, and model accuracy were collected and analyzed.

6. **Feedback and Iterative Refinement**: Initial testing and demonstrations provided valuable feedback, particularly from domain experts and potential end-users. This feedback informed iterative adjustments to improve user experience, interface design, and feature functionality. The continuous integration of user feedback ensures that the frameworks are refined to meet diverse needs in real-world scenarios.

By combining literature analysis, structured framework design, practical prototyping, and performance evaluation, this methodology provides a comprehensive approach to creating, implementing, and validating DSS frameworks that enhance decision-making processes, data quality, and user accessibility.

1.5 Thesis Contributions

This thesis aims to provide significant advancements in AI-based DSS by introducing frameworks and modules that improve accessibility, usability, and trustworthiness of AI technologies. In this section, the main contributions are discussed.

- **Novel DQM framework for DSS.** The DQM framework enhances the pre-processing phase of AI models by providing detailed assessments of data quality and ease of cleaning. Adhering to European Union (EU) guidelines for AI, the module ensures that datasets meet high-quality standards, thereby improving the reliability and performance of AI models. This innovation addresses critical challenges in data preparation, which is a foundational stage in AI pipeline development.

- **Simplified AI for non-experts.** DSS4EX is a user-friendly framework designed to democratize AI by enabling non-experts to better understand and manipulate AI pipelines. Through an intuitive interface and embedded educational tools, it simplifies complex AI processes, reduces barriers to adoption, and fosters broader engagement with AI technologies across various domains.
- **Comprehensive AI pipeline support.** The EX-DSS framework integrates the DQM and DSS4EX into a comprehensive system that supports the entire lifecycle of AI pipeline design and deployment. With enhanced flexibility, advanced customization options, and user guidance features, EX-DSS allows users to tailor AI pipelines to their specific needs, facilitating efficiency and innovation without requiring deep technical expertise.
- **Practical applications in real-world scenarios.** The proposed frameworks have been applied to practical scenarios, such as smart plug energy forecasting, which improves energy efficiency in commercial buildings. These applications demonstrate the frameworks' utility in solving real-world challenges and delivering impactful outcomes.
- **Promoting trust and understanding in AI.** Embedded tools within DSS4EX and EX-DSS promote user understanding of AI systems, fostering transparency and trust. By enabling users to interact with and refine AI pipelines while providing contextual explanations, these tools build confidence in AI systems and encourage their responsible and widespread use.

Figure 1.3 provides a high-level overview of the DSS architecture proposed in this thesis. This structure is composed of two primary modules, Data Quality (DQ) and Machine Learning (ML), designed to integrate expert knowledge and user feedback within the decision-making process. The first module, the DQM, is responsible for assessing and improving the quality of user input, thereby supporting effective resource management and ensuring that the data used in the system meets the required standards (Chapter 2).

The second module, focused on AI pipeline design, offers two configuration paths: a structured, predefined setup (Chapter 3) and a more flexible, user-guided structure (Chapter 4). The "Human-in-the-loop" component, represented by a parallelogram, emphasizes the role of the expert in guiding configuration and providing knowledge-based suggestions that enhance the AI pipeline's relevance and accuracy. This module enables the user to interactively configure, design, and refine the pipeline, benefiting from system-provided explanations and contextual guidance at each stage.

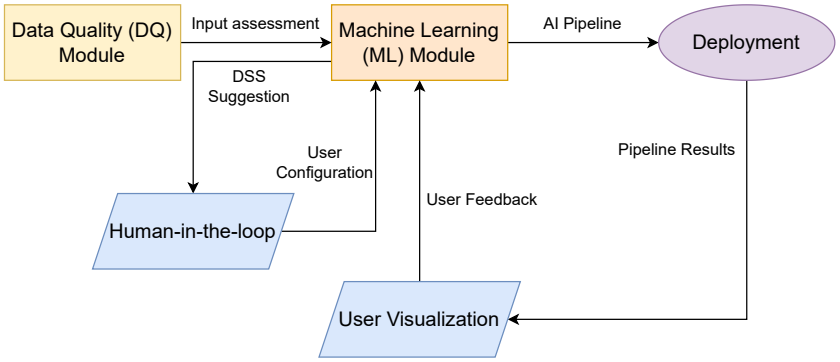


Figure 1.3. **Overview of the Decision Support System (DSS) Architecture Proposed in this thesis.** This high-level diagram illustrates the modules that constitute the DSS framework developed in this research. The rectangles represent the two primary modules: the Data Quality Module (DQM), which assesses and enhances data quality, and the Machine Learning (ML) module, which is responsible for managing AI pipeline configuration and deployment. The "Human-in-the-loop" parallelogram signifies the expert's involvement during configuration, where human input not only directs the setup but also provides expert knowledge-driven suggestions to guide decision-making. The ellipses represent key outputs: the deployment phase of the AI pipeline and the resulting outputs from the ML module, which are presented to the user through an interactive visualization interface. This design supports a collaborative process where user feedback refines system outputs, enhancing the DSS's responsiveness and alignment with expert insights.

Once the pipeline configuration is complete, the system initiates deployment, visualized through an ellipsis that represents the deployment and resulting output. The AI pipeline output is then delivered to the user via an intuitive visualization interface. The system supports a collaborative process, as users can retain results for future reference or provide feedback, which further refines and enhances system performance. This feedback loop reinforces the DSS's adaptability and responsiveness, aligning it more closely with expert insights and real-world application requirements.

Figure 1.4 provides a structured visualization of the interconnected components and relationships explored in this thesis. Organized into a three-column structure, the figure illustrates the progression of the research. The first column highlights the key contributions, offering a high-level summary of the innovative advancements introduced. Arrows connect these contributions to the second column, where each is thoroughly expanded and discussed in specific chapters. The third column features the primary modules, the DQM and the ML module, which constitute the core of the DSS. These modules are directly linked to the corresponding chapters for detailed explanations. Furthermore, the supplementary modules depicted in Figure 1.3 are elaborated

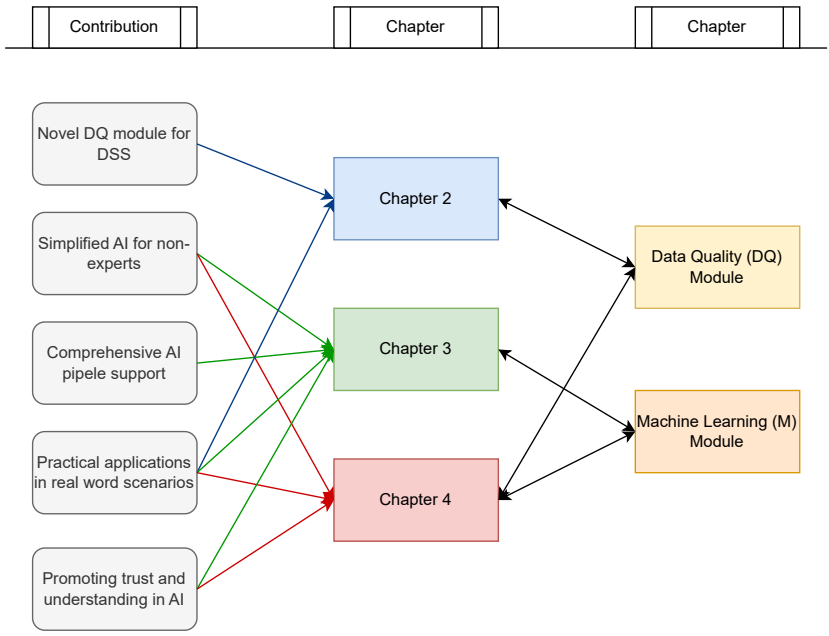


Figure 1.4. **Structured overview of the Thesis components.** This visualization illustrates the relationships among the key contributions, chapters, and main modules of the Thesis. The first column summarizes the contributions, while arrows link them to the chapters (second column) where they are discussed in greater detail. The third column highlights the core modules of the DSS, including the Data Quality (DQ) and Machine Learning (ML) modules, along with their connections to the respective chapters.

upon within their respective chapters, providing a comprehensive understanding of the framework's architecture and functionality.

1.6 Thesis Scope and Limitations

Even though this thesis presents significant advancements in AI-based DSS, it is important to recognize that the frameworks introduced are still in their prototype phase. This study focuses on developing and integrating AI-driven modules that enhance data quality and improve the accessibility and usability of AI technologies for a broader user base. The core contributions include the design of the DQM, which ensures the quality and trustworthiness of data during preprocessing, and DSS4EX. The latter enables non-experts to explore

and configure AI pipelines. These components are integrated into the EX-DSS system, providing comprehensive support throughout the AI pipeline lifecycle, from design to deployment.

The frameworks are demonstrated in a real-world use case within the energy efficiency domain, specifically for smart plug energy forecasting in commercial buildings. While the study's primary focus is on the energy sector, the modular design of the frameworks allows for potential applications in other areas, such as healthcare, finance, and logistics. However, the prototypes have only been tested in this specific domain, which limits their adaptability to other sectors. Future research could expand testing to different data types and decision-making contexts to assess the broader applicability of the frameworks.

Additionally, as prototype systems, EX-DSS and DSS4EX require further refinement for production deployment, particularly in terms of robustness, scalability, and usability. Although the current versions effectively demonstrate the core functionalities, transitioning them to fully operational systems will require more development. Furthermore, while the frameworks currently support batch data processing, they are not yet capable of real-time data processing. Expanding their capabilities to handle real-time data would enhance their applicability in dynamic environments, such as autonomous systems, real-time decision-making, or healthcare monitoring.

Another area for future exploration is the integration of the frameworks with existing DSS and AI tools. Ensuring compatibility with established workflows could increase their usefulness for organizations with existing infrastructures. Lastly, while privacy and security are acknowledged, they were not fully addressed in this research. Incorporating privacy-preserving mechanisms and stronger security measures will be crucial, especially when applying the frameworks in sensitive domains like healthcare or finance. Addressing these limitations will improve the frameworks' versatility and broaden their impact across various real-world applications.

1.7 Structure Thesis

Chapter 2: DQM - Data Quality Module for Decision Support Systems
This chapter explores the critical role of data quality in the effectiveness of decision support systems. It presents a detailed framework for the Data Quality Module (DQM), which assesses and improves the qualitative level of the data using guidelines such as those from the EU. The chapter also highlights the module's impact on the pre-processing stage of AI models, ensuring better data integrity and performance in downstream AI processes.

This chapter is detailed in the following paper:

Rinaldi, Giulia; Crema Garcia, Fernando; Agudelo, Oscar Mauricio; Becker, Thijs; Vanthournout, Koen; De Moor, Bart
"A Framework for a Data Quality Module in Decision Support Systems: an application with Smart Grid Time series"
ICEIS 2023 - The 25th International Conference on Enterprise Information Systems, Prague (Czech Republic), April 2023
DOI: 10.5220/0011749700003467
https://www.scitepress.org/Link.aspx?doi=10.5220/0011749700003467

Chapter 3: DSS4EX – A Framework for Exploring Artificial Intelligence Pipelines
This chapter introduces the DSS4EX framework to make AI pipelines more accessible to non-experts. It explains how the framework simplifies the exploration and understanding of AI models by providing an intuitive interface and educational tools. The chapter includes both technical details and practical examples of how DSS4EX can be used to navigate complex AI pipelines, making AI technologies more user-friendly and approachable.

This chapter is detailed in the following paper:

Rinaldi, Giulia; Theodorakos Konstantinos; Crema Garcia, Fernando; Agudelo, Oscar Mauricio; De Moor, Bart
"DSS4EX: A Decision Support System Framework to Explore Artificial Intelligence Pipelines with an application in Time Series Forecasting".
Expert Systems with Applications, ISSN: 0957-4174,2025
DOI: 10.1016/j.eswa.2025.126421
https://www.sciencedirect.com/science/article/pii/S0957417425000430

Chapter 4: EX-DSS – An Explorative Decision Support System
Building on the earlier modules, this chapter presents EX-DSS, a comprehensive framework that integrates both the DQM and DSS4EX. It discusses how EX-DSS supports the full lifecycle of AI pipeline design, from initial input to deployment. A significant focus is placed on its real-world application in smart plug forecasting, demonstrating how EX-DSS enhances energy efficiency and performance in commercial buildings.

This chapter is detailed in the following paper:

Rinaldi, Giulia; Botman, Lola; Agudelo, Oscar Mauricio; De Moor, Bart
"EX-DSS: An Explorative Decision Support System for designing and deploying Smart Plug Forecasting Pipelines"
KMIS 2024 - The 16th International Conference on Knowledge Management and Information Systems, Porto (Portugal), November 2024
DOI: 10.5220/0012925800003838.
https://www.scitepress.org/Papers/2024/129258/129258.pdf

Chapter 5: Conclusion and Future Directions
The final chapter summarizes the key contributions and findings of the Thesis. It provides a reflection on the broader implications of the research, including how the proposed frameworks contribute to the democratization of AI-driven decision support systems. Additionally, the chapter addresses the limitations encountered during the research and offers recommendations for future studies, particularly in improving customization features and exploring other real-world applications of the developed systems.

1.8 Acknowledgments

This research received funding from

- KU Leuven: Research Fund (projects iBOF/23/064, C3/20/117, C3I-21-00316), Industrial Research Fund and several Leuven Research and Development bilateral industrial projects;
- Flemish Government Agencies:
  - FWO: SBO project S005319N, TBM Project T001919N; FWO PhD Grant 11K5623N;
  - EWI: the Flanders AI Research Program;
  - VLAIO: CSBO (HBC.2021.0076) Baekeland PhD (HBC.20192204);
- EU: This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 885682). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect

those of the European Union or ERC. Neither the European Union nor the granting authority can be held responsible for them.

The work presented in Chapter 2 was conducted in collaboration with Stadius, VITO, and Fluvius as part of the Flanders AI Research Program.

The work presented in Chapters 3 and 4 was carried out within the research group of Stadius.

I did use the generative AI assistance tools, such as ChatGPT, Grammarly, and DALL-E.
Generative AI tools were used to generate the cover image and to improve language and clarity in the writing of this thesis.
The text/code/ images in this thesis are my own (unless otherwise specified) and generative AI has only been used in accordance with the KU Leuven guidelines and appropriate references have been added. I have reviewed and edited the content as needed and I take full responsibility for the content of the thesis.

Chapter 2

DQM: Data Quality Module for Decision Support System

This chapter is a published paper which has been modified and reprinted with permission. The original version can be found on:

Rinaldi, Giulia; Crema Garcia, Fernando; Agudelo, Oscar Mauricio; Becker, Thijs; Vanthournout, Koen; De Moor, Bart
"A Framework for a Data Quality Module in Decision Support Systems: an application with Smart Grid Time series"
ICEIS 2023 - The 25th International Conference on Enterprise Information Systems, Prague (Czech Republic), April 2023
DOI: 10.5220/0011749700003467
https://www.scitepress.org/Link.aspx?doi=10.5220/0011749700003467

Abstract

Data Quality (DQ) measures data status based on different dimensions. This broad topic was brought to the fore in the '80s when it was first discussed and studied. A high-quality dataset correlates with good performance in Artificial Intelligence (AI) algorithms and decision-making processes. Therefore, checking the quality of the data inside a Decision Support System (DSS) is an essential preprocessing step and is beneficial for improving further analysis. This chapter presents a theoretical framework, referred to as DQM, for a DQ module in a DSS. The framework evaluates the quality status in three stages: checking that data complies with European guidelines, measuring the data using the DQ metrics criteria, and running a process to detect common problems in Data Cleaning (DC). Additionally, the DQM framework supports the user in identifying and fixing the DC problems, which speeds up the process. As output, the user receives a DQ report and the DC pipeline to execute to improve the dataset's quality. An implementation of the framework is illustrated in a Proof-Of-Concept (POC) for an industrial use case. In the POC, an example of the execution of the various framework phases is shown using a public time series dataset containing quarter-hourly consumption profiles of residential electricity customers in Belgium for the year 2016.

2.1 Introduction

For most Artificial Intelligence (AI) projects, the first step an analyst needs to perform is to evaluate the quality of the received data. Following this evaluation, the analyst conducts a detailed examination of the data to gain a comprehensive understanding of its characteristics and potential issues. This investigative process allows the analyst to assess Data Quality (DQ) and identify areas for enhancement. If necessary, appropriate measures are implemented to improve DQ before proceeding with further analysis. Therefore, DQ measures the state of the data based on various factors, including accuracy, completeness, consistency, and timeliness. According to Ehrlinger et al. (2019) [28], these four DQ metrics are considered the most fundamental. Consequently, the research presented in this chapter will focus on them. The process of fixing the possible dataset issues is called Data Cleaning (DC), which is composed of preprocessing routines fundamental to guaranteeing the success of further analysis. For instance, it was demonstrated that DQ influences the error rate of machine learning models [28] and the decision-making process [19].

Appen, an AI company, submits yearly surveys among data scientists with questions related to Machine Learning (ML) models and data to investigate the state of AI and ML. In the 2018 report, most participants determined that the quality of the training data was their biggest challenge.[1] In 2022, more than 70% of participants declared that they spent at least 30% of their time on preprocessing tasks [2]. These findings highlight the significant time and effort required for data preparation. Therefore, assisting analysts during DQ evaluation and DC processes could substantially improve research efficiency.

Assisting an analyst during this first phase can help to speed up a critical process that requires time and attention. Common practice reveals the tendency of analysts to do their own custom DC process from scratch, even in cases when this had previously been done before by another analyst. This repetition of work could be seen as a waste of resources. Organizing the job in a way that researchers can evaluate and apply the same DC process to the datasets would allow them to save time.

This chapter aims to minimize the need for repeated DC routines and enhance the efficiency of DQ assessment. To achieve this, it introduces the DQM framework, which provides a structured approach for handling DQ within a Decision Support System (DSS). A DSS is a computer software tool that aids a user in managing and making decisions on the data. The defined module

[1] https://visit.figure-eight.com/rs/416-ZBE-142/images/Data-Scientist-Report.pdf

[2] https://appen.com/blog/2022-state-of-ai-machine-learning-report/

analyzes the input to extract useful information during the data-cleaning process. It checks the quality of the inputs in three processes:

- EU guidelines process: the focus is mainly on the metadata and reusability of the dataset.
- DQ assessment process: data quality dimensions are checked on the dataset.
- DC assessment process: The system evaluates some common data issues.

As a result of performing the three analyses through the DSS, the system computes two indicators that summarize the last two processes. The indicators and the analysis details are shown in a final report to submit to the analyst. In the last phase of the DQM for DSS, the system assists the user in designing the DC process by highlighting some cleaning problems using the information gathered during the previous analysis and using historical experiences stored in a database. The expert decides how to handle them, and each feedback given is saved as historical experience to be available for future analysis on future datasets. Therefore, the framework intends to offer the analyst a way to study the dataset while simultaneously speeding up the preprocessing design pipeline. The framework is then tested on a public dataset, unlike the usual approach, which focuses on well-known organizational settings and assumes private or familiar datasets with available metadata [82]. In this thesis, we consider an open dataset to demonstrate that the proposed framework can be applied to any kind of dataset, regardless of its origin or structure.

In order to elaborate further the proposal of the DQM, Section 2.2 presents some of the most pertinent research on frameworks and tools for data quality and cleaning. Section 2.3 provides an overview of the research objectives and discusses the challenges addressed in this work. Section 2.4 explains our framework for a data quality module to be used inside a DSS. Section 2.5 illustrates this framework when applied to a use case. Section 2.6 draws conclusions and delineates possible further research directions.

2.2 Related Work

DQ began to gain prominence in the 1980s, as organizations increasingly relied on data and data mining for decision-making and business operations. Many dimensions and metrics were defined during the years in various studies and then translated into practical software tools. Ehrlinger et al. (2022) [30] identified 667 tools developed for DQ tasks. The authors underlined the difficulty of

identifying just one definition of DQ. The main contributions of their work include data profiling, i.e. the process of collecting metadata; data quality measurements, i.e. the capability of estimating DQ dimensions; and data cleansing, i.e. the procedure for fixing incorrect data. To evaluate the 667 tools, a requirements catalog consisting of 43 constraints for data profiling, data quality measurements, and continuous data quality monitoring was defined. Only 13 tools respected the constraints and were then described in detail. The authors concluded that there is still a need to research automation in DQ, and the examined tools missed a clear *"declaration and explanation of the performed calculation and algorithms"*.

Ehrlinger et al. (2017) [29] proposed an automated data quality monitoring tool, which periodically monitors the data quality of heterogeneous data collected by an information system. The employed architecture is comprised of four components. The first is data profiling and quality assessment. During this phase, the metadata and calculated DQ metrics are collected. The user can also provide domain-specific information and additional specification used in the process, such as the monitoring frequency. The results calculated over time are stored in the DQ repository, the second component. The third element is time series analysis which integrates commonly used algorithms, such as Cumulative Sum (CUSUM) methods, Exponentially Weighted Moving Average (EWMA) and Autoregressive Integrated Moving Average (ARIMA) models, to examine the information in the DQ repository. The last component is visualization, in which the user can plot the collected time series and monitor the obtained results.

The framework proposed by Oliveira et al. (2022) [67] monitors DQ using a reliability score. The input of the system is heterogeneous data. Their architecture is based on the scalable publish/subscribe messaging system, Kafka (`https://kafka.apache.org/`). The main component is the data quality layer which is composed of a plug-in and the data quality analyzer. The latter is the process that produces a data quality index employing rules to analyze the data stored in a JSON (JavaScript Object Notation) file. The plug-in applies and checks the rules on the data. The authors tested the framework to a use case using customer data. In the example, a reliability score was calculated using the DQ dimensions of accuracy, consistency, and completeness. It was used to identify possible outliers.

Like the papers described in this section, the DQM framework intends to monitor the quality status of the data. The main difference is that the DQM framework offers three levels of assessment. The first focuses on the input's format and standard, the second calculates DQ against a set criteria, and the third assesses common DC problems. As output, the user will have three evaluations to estimate how much time is needed to clean a specific dataset.

The tracking of data cleaning changes represents a novel contribution within the DQM framework. This feature records the transformations applied to improve data quality, preserving them as valuable historical insights for future analyses of similar datasets. In this way, the system learns from the user and speeds up future processes by offering better support. The system identifies the cleaning problems and solves them with the human-in-the-loop approach. At the end, the user is provided with a report of the raw data and the preprocessing pipeline.

2.3 Chapter Objectives and Challenges

This section introduces the challenges of DQ and DC in DSS, such as redundant workflows, limited reuse of past solutions, and poor integration of human expertise with automated tools. To address these issues, it presents a novel Data Quality Module (DQM) framework that combines automated assessments with expert feedback, reuses historical cleaning processes, and provides detailed reports to support decision-making. The framework's design was illustrated through a POC implementation based on a publicly available industrial Belgian smart grid dataset.

- **Problem Statement: Importance of DQ and DC**:
  - Literature in AI and ML consistently highlights the critical role of DQ and DC in ensuring reliable and effective applications.
  - Despite advancements, several challenges remain unresolved, particularly in improving efficiency, reproducibility, and accessibility in DSS data preparation.
- **Key Challenges in Current Systems**:
  - **Redundant cleaning workflows**: Analysts often create custom cleaning pipelines for each project, leading to inefficiencies.
  - **Lack of learning from past processes**: Existing systems do not leverage insights from previous DC efforts, missing opportunities for reusable solutions.
  - **Poor integration of user expertise with automation**: Most tools fail to effectively combine automated DQ assessments with human input, limiting adaptability for diverse datasets.
  - **Inadequate reporting mechanisms**: Reports often lack clarity, making it difficult for analysts to quickly assess data quality and readiness.

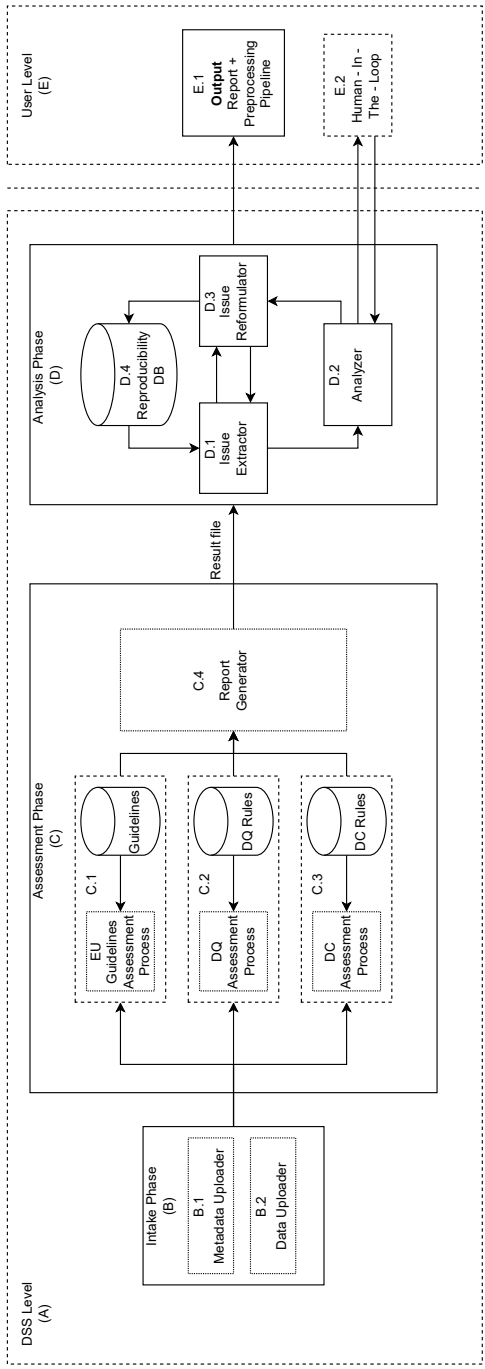


Figure 2.1. **Framework architecture overview.** The diagram illustrates the two main concept levels of the framework: the user level (E) and the DSS level (A), along with the key phases involved in the data processing pipeline. The DSS level includes the Intake phase (B), Assessment phase (C), and Analysis phase (D), detailing the subphases and interactions across each stage, from metadata and data upload to the generation of reports and cleaning procedures.

- **Proposed Solution: A Novel DQM for DSS**
  - Introduces a comprehensive data quality evaluation approach, ensuring datasets meet high standards by assessing:
    - Compliance with established guidelines.
    - Key data quality metrics.
    - Common data cleaning issues.
  - Incorporates a historical repository of transformations and feedback:
    - Enables reusable solutions.
    - Minimizes redundancy and optimizes workflows.
  - Integrates human-in-the-loop functionality:
    - Bridges automation with expert decision-making.
    - Enhances flexibility and precision in handling data issues.
  - Provides detailed reports summarizing:
    - DQ assessments.
    - Suggested preprocessing strategies.
  - Streamlines the analysis process for better decision-making.
- **Validation Through Real-World Application**
  - The proposed DQM framework is applied to a real-world use case involving a Belgian smart grid dataset.
  - Demonstrates the system's practical utility and adaptability in handling real-world data.
- **Impact and Contribution**
  - Advances the state of DQ and DC by shifting from inefficient, repetitive practices to a structured, reusable data preparation approach.
  - Promotes collaboration, reproducibility, and accessibility in DSS and AI workflows.
  - Ensures DSS and AI applications become more practical, scalable, and accessible to a broader audience.

2.4 The DQM Framework

The architecture of the DQM framework is shown in Figure 2.1, consisting of two conceptual levels: the user and DSS levels. The user level (Fig. 2.1E)

represents the client, while the DSS level (Fig. 2.1A) operates on the server side.

The DSS level begins with the Intake phase (Fig. 2.1B), and it describes how the user should upload the metadata and the data into the DSS. There are two subphases: metadata uploader (Fig. 2.1B.1) and data uploader (Fig. 2.1B.2). The first aims to define the metadata input related to the data, which should include the description and source of the data. On the other hand, the data uploader determines the architecture constraints such as the data format – for example, CSV (Comma Separated Variable) file for tabular and time series data or tiff (Tag Image File Format) for images. In the Proof-Of-Concept (POC) described in Section 2.5, this last option is not implemented but is part of future research. The metadata and the data are the input for the next phase, the assessment phase (Fig. 2.1C), which evaluates the user's input in three stages composed of an assessment process and a database containing rules or useful information.

The first stage is the EU guidelines assessment process (Fig. 2.1C.1), which uses rules taken from the European Guidelines [22]. The European data quality guidelines are recommendations from the European Union to produce high-quality datasets. Their proposal uses the principles of FAIR [110], which stands for four data quality dimensions: Findability, Accessibility, Interoperability, and Reusability. The guidelines provide a framework consisting of these dimensions and additional metrics. The EU guidelines assessment process interacts with a database where the rules extracted from the guidelines are stored. The submitted metadata should follow a template defined during the setup of the DSS. The EU guidelines assessment process follows the dimensions:

- Interoperability: The EU guidelines assessment process verifies any encoding issues in reading metadata or opening the data files.
- Findability: The EU guidelines assessment process checks if the metadata at least contains the description of the data, the source of the data, and if it is part of a project, the description of the project. It measures how easy it is to find information and understand the data.
- Accessibility: The EU guidelines assessment process checks how many files the data are divided into and if they are all accessible. Then, it verifies if there are constraints, for example, security constraints, to maintain inside the DSS, such as, for example, "only the admin can access the data."
- Reusability: The EU guidelines assessment process checks the amount of data and any additional rules related to the specific file type. For example, if the file is a CSV, the DSS should check the presence of headers. If the

organization using the DSS requires any supplementary standard to be respected, the system will review them at this point.

The result of each phase is saved and shown to the user in a final report.

The second stage of the assessment phase is the DQ assessment process (Fig. 2.1C.2). The rules related to this stage refer to the data quality metrics:

- Accuracy: The DQ assessment process measures how correct the data are compared to a reference dataset.
- Timeliness: The DQ assessment process evaluates how up-to-date the input dataset is for a task.
- Completeness: The DQ assessment process measures how much information the data carries out. In other words, how much data is not missing from the dataset.
- Consistency: Semantic rules are defined over the data by the user. The DQ assessment process estimates the number of semantic rules which are violated.

Based on the results, a DQ indicator is calculated. This number establishes the data's quality status at any moment. The intent is to provide a summary of the status of the data that is simple to understand but also possible to change depending on the context of the problem. The next steps are a possible guide to calculate this indicator:

1. Calculate quality indicators DQ_i per each dimension *i= {comp, acc, time, cons}* completeness, accuracy, timeliness, and consistency. Each indicator will be defined by the analyst.
2. Assign importance weights (w_i) per each of the data quality dimensions.
   - If the units of DQ_i have the same range, we recommend updating each w_i as:

$$w_i = \frac{w_i}{\sum_k w_k} \text{ so } \sum_i w_i = 1 \implies \overline{DQ} = \sum_i w_i DQ_i \qquad (2.1)$$

   - If the analyst wants to do the average of the DQ_i, update each w_i as:

$$w_i = 1 \text{ so } \sum_i w_i = 4 \implies \overline{DQ} = \frac{\sum_i DQ_i}{4} \qquad (2.2)$$

3. Calculate a weighted average of each data quality indicator that we denote as $\overline{DQ}$.

$$\overline{DQ} = \frac{\sum_i w_i DQ_i}{\sum_i w_i} \tag{2.3}$$

4. Then, compare this value to a threshold ϵ with a basic rule:

$$\textbf{if } \overline{DQ} \leq \epsilon \textit{ insufficient quality } \textbf{else } \textit{sufficient quality}$$

   This step is recommended but optional.

5. Add results to the result file.

Section 2.5 presents an example of the DQ indicator.

The final stage of the assessment phase is the DC assessment process (Fig. 2.1C.3). The rules are based on typical instructions during the data-cleaning procedure. These include

- Time column: The DC assessment process tries to recognize the time column, the start, and the end date, the frequency of the observations, and the presence of gaps or duplicates related to winter/summer hour time change.
- Single-value columns: The DC assessment process verifies how many columns have only a single value. It should be stored as metadata.
- Types of columns: The DC assessment process tries to assign a specific type to a column based on the values.
- Duplicates: The DC assessment process extracts possible duplicates.
- Missing Values: The DC assessment process verifies if there are any missing values.

As with the previous DQ stage, a DC indicator is calculated based on the result of this stage too. This number aims to provide an idea of the cleaning status of the data. It measures how easily the DSS can clean this dataset without human intervention. This information is linked to the analysis phase because the less information the system defines, the more time is required during the following phase. Under this point of view, the analysts should use it to estimate the time they will probably need to spend during the analysis phase when additional verification will be done. Section 2.5 illustrates an example of the indicator's computation. The modularity of the DQM framework offers flexibility. The analyst can decide to skip part of the analysis and can update or change the

rules and guidelines to make the DQM more customizable. After the three assessment stages, the last step in the assessment phase generates a report (Fig. 2.1C.4) summarizing all the DSS findings.

The last part of the DQM framework is the analysis phase (Fig. 2.1D) which aims to define the cleaning procedure with the analyst and save it in the reproducibility database. The phase starts with the issue extractor (Fig. 2.1D.1), a process that formulates potential problems based on the assessment phase report and historical experience already stored in the reproducibility database (Fig. 2.1D.4). Then, these problems are displayed to the analyst through the analyzer (Fig. 2.1D.2), which explains the potential issues and shows additional information, such as plots or metadata. At this point, the analyst (Fig. 2.1E.2) decides whether or not the issue is legitimate and decides on a solution. Finally, the analyst's feedback is passed to the issue reformulator (Fig. 2.1D.3) and stored in the reproducibility database. This database contains the past problems found by the DSS and how analysts have decided to solve them. It helps the user verify the most common issues and situations that may not be so ordinary. The output of the DQM framework (Fig. 2.1E.1) is a report on the raw dataset and the preprocessing pipeline designed during the analysis phase.

2.5 An industrial application of DQM at a proof-of-concept level.

This section presents the design and key characteristics of DQM for a DSS within a Proof-Of-Concept (POC) implementation. It serves as the prototype's documentation, detailing its objectives, architecture, requirements, and implementation choices. This structured analysis provides insight into the module's functionality and design principles.

A research-driven prototype was implemented to demonstrate the proposed DQM framework's feasibility and integration in a DSS context. It integrates automated assessments with human-in-the-loop decision-making to enhance flexibility and adaptability. Designed with a modular approach, it assesses data quality based on established guidelines, predefined metrics, and common data cleaning challenges. Additionally, by leveraging a historical repository of past transformations, the system enhances reproducibility and reduces redundancy in data preprocessing workflows.

The following Subsections outline the core aspects of the prototype, including its scope, functional and non-functional requirements, system architecture, and user interactions. Furthermore, testing and validation strategies are described

Type	InstallatieID	Afname/Injectie	Meter read tijdstip	Eenheid van meetwaarde	Meetwaarde	Status
Elektriciteit	K6LUKJZ1BtSijA	Afname	01JAN16:16:00:00	Kwh	0,9800	VAL
Elektriciteit	K6LUKJZ1BtSijA	Afname	01JAN16:16:15:00	Kwh	0,9800	VAL
Elektriciteit	K6LUKJZ1BtSijA	Afname	01JAN16:16:30:00	Kwh	0,6600	VAL
Elektriciteit	K6LUKJZ1BtSijA	Afname	01JAN16:23:00:00	Kwh	1,0000	VAL
Elektriciteit	K6LUKJZ1BtSijA	Afname	01JAN16:23:15:00	Kwh	1,0000	VAL

Figure 2.2. **Example dataset used in the proof-of-concept.** This CSV file contains quarter-hourly offtake and injection measurements for Low-Voltage (LV) grid connections in Belgium. The data gathered from a pilot project conducted in 2016 consists of anonymized readings with over 98% validation. The dataset includes 100 time series of validated measurements from digital meters in a specific Belgian area. Key columns include: `'Type'` (measurement type), `'InstallatieID'` (digital meter ID), `'Afname/Injectie'` (off-take or injection of energy), `'Meter read tijdstip'` (timestamps), `'Eenheid van meetwaarde'` (unit of measurement), `'Meetwaarde'` (measurement value), and `'Status'` (data validity).

to ensure that the module aligns with its intended research and analytical objectives.

2.5.1 General Description

- **Software Prototype Title**: Data Quality Module for Decision Support System
- **Overview**: The Data Quality Module (DQM) for a Decision Support System (DSS) is designed to evaluate and improve the quality of data used in decision-making processes. The module implements a structured data assessment and cleaning approach in three key phases:

  1. **EU Guidelines Compliance**: Evaluates metadata according to European data quality recommendations.
  2. **Data Quality Assessment**: Measures key dimensions such as accuracy, completeness, consistency, and timeliness.
  3. **Data Cleaning and Enhancement**: Detects and suggests solutions for common data issues, improving dataset usability.

  By providing automated assessments and recommendations, the module enhances the efficiency and reliability of downstream data analysis.

- **Objectives**:
  - **General Purpose**: The module assists users in identifying and resolving data quality problems while generating comprehensive reports on the dataset's quality status.
  - **Goals**: The main goals of the DSS Data Quality Module are:
    - **Evaluate** data quality based on EU guidelines, established data quality metrics, and typical data cleaning issues.
    - **Provide** clear indicators summarizing dataset quality.
    - **Assist** users in detecting and resolving data cleaning issues.
    - **Generate** structured reports and a proposed data cleaning pipeline.
- **Scope**:
  - **Targeted Functionalities**: The system covers the following functionalities.
    - Metadata and data quality evaluation based on EU guidelines.

  - Assessment of data quality dimensions: accuracy, completeness, consistency, and timeliness.
  - Identification and suggestion of solutions for common data cleaning issues.
  - Integration with a reproducibility database for leveraging historical cleaning records.
  - Generation of detailed reports summarizing data quality and cleaning recommendations.
- **Excluded Functionalities**: The following functionalities are not included in the current prototype because they have not been regarded as essential for the data quality assessment.
  - User authentication and role-based access control.
  - Development of advanced machine learning models.
  - Integration with external project management tools.
  - Support for offline mode operation.

- **System's Requirements**
  - **Functional Requirements**: In order to effectively run the framework's processes, the chosen system must provide the following functionalities.
    - **Data and Metadata Upload**: Users can upload metadata and datasets (supported formats: CSV, JSON).
    - **Data Quality Evaluation**: The system must assess dataset quality using predefined EU guidelines.
    - **Quality Metrics Computation**: The system evaluates accuracy, completeness, consistency, and timeliness.
    - **Data Cleaning Assessment**: The system must automatically identify:
      - Time columns and their structure.
      - Single-value columns.
      - Data type inconsistencies.
      - Duplicates and missing values.
    - **Report Generation**: The system must generate structured reports summarizing the data quality assessment and cleaning suggestions.
    - **Historical Learning Integration**: The system must leverage a reproducibility database to recommend cleaning strategies based on previous user actions.

  - **User Interaction**: Users must be able to review and confirm identified issues, adjust suggested cleaning steps, and save feedback for future reference.
- **Non-Functional Requirements**: In addition to functional specifications, the system must meet the following non-functional requirements:
  - **Performance**: The system should provide timely feedback, ensuring dataset processing occurs within an acceptable timeframe.
  - **Security**: User data must be securely stored and accessed according to best practices.
  - **Usability**: The user interface was designed with intuitiveness and user-friendliness in mind, although no formal usability testing was conducted.
  - **Scalability**: The system should handle large datasets and support integration with multiple data formats.
  - **Compatibility**: The system should be compatible with web browsers and common data file formats.

- **Architecture**: The architecture of the DQM for DSS follows a **client-server model**, where a centralized DSS web server manages data processing and interacts with multiple clients. The system consists of two main components:
  - **Client Side**: Users interact with the system through a web-based interface built with Streamlit. The client interface allows data upload, configuration of constraints, and visualization of reports.
  - **Server Side**: The backend, implemented in Python, handles data ingestion, quality assessment, issue detection, and report generation. The DSS web server processes user inputs and returns structured feedback.

- **Software Network-based Architecture**: Figure 2.3 illustrates the network-based architecture. The DSS web server acts as the central processing unit, serving multiple clients. Users communicate with the DSS via HTTP requests, submitting datasets for analysis and retrieving quality reports.
- **Processing Flow**: The DSS module follows a structured data processing flow divided into three main phases. Figure 2.4 illustrates the logical flow of user interaction with the DSS.
  1. **Intake Phase**: Users initialize a new project and upload metadata and datasets.

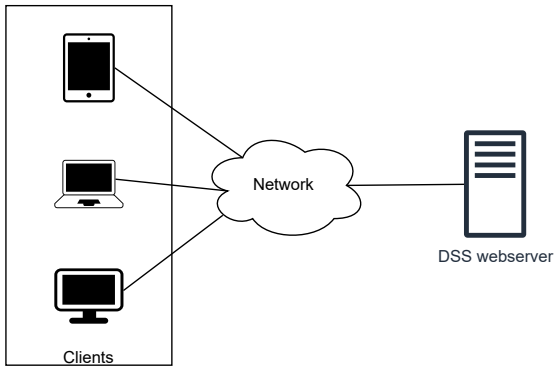


Figure 2.3. **Network Architecture of the DSS.** The DSS follows a client-server paradigm, where multiple client applications communicate with a centralized DSS web server over a network. The web server handles requests for data quality analysis and sends responses to clients.

2. **Assessment Phase**: The DSS evaluates data quality based on predefined metrics and EU guidelines.
3. **Analysis Phase**: Identified issues are analyzed, potential historical cleaning solutions are retrieved, and a final preprocessing pipeline is proposed.

Each phase integrates user feedback, enabling interactive decision-making. The final output consists of a structured data quality report and a cleaning pipeline recommendation, ensuring enhanced data reliability for downstream analysis.

- **Main User Stories**: Below, there are 2 user stories where target clients describe the actions that they need the software to perform.
  - **User Story 1**: *As a Data Analyst, I want to evaluate the dataset's quality so that I can detect and address potential issues before performing detailed analysis.*
    - **Acceptance Criteria**:
      - The system must allow users to upload metadata and dataset files.
      - The system must generate a comprehensive data quality report.

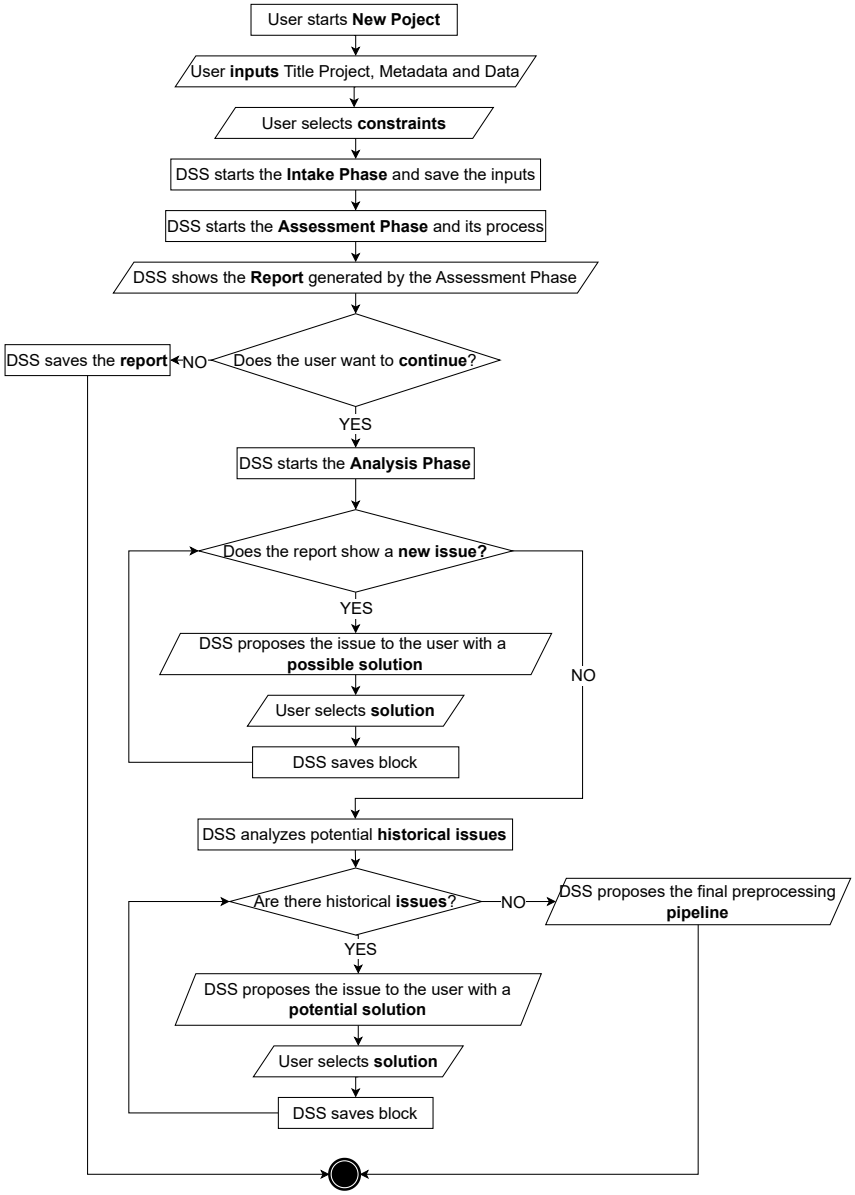


Figure 2.4. **Data Processing Flow of the DSS.** This diagram outlines the sequential phases of the DSS: data intake, quality assessment, and interactive analysis. Users provide inputs, receive feedback, and refine the dataset through an iterative process.

- **User Story 2**: *As a Data Analyst, I want to identify and fix data cleaning issues so that I can ensure high-quality data for decision-making.*
  - **Acceptance Criteria**:
    - The system must identify and suggest solutions for common data cleaning problems.
    - The system must integrate user feedback and historical cleaning experiences.

2.5.2 DQM Real-life implementation at a proof-of-concept level

This subsection provides an in-depth analysis of the processes incorporated into the POC implementation of DQM for a data-driven DSS. The POC was developed for a research organization and tested using a public industrial time series dataset published by Fluvius, a Belgian distribution system operator. The dataset consists of quarter-hourly electricity consumption profiles from residential customers in Belgium for the 2016 calendar year. The use case focuses on leveraging AI techniques to improve the management of low-voltage grids, as part of the Smart Grid use case within the AI Flanders Research Program[3].

Building on the previously outlined generalities, objectives, scope, requirements, architecture, user interactions, and validation strategies, this Section explores the specific processes implemented within the prototype. It details each step of the software workflow, including data intake, quality assessment, issue detection, cleaning recommendations, and reporting. These insights aim to provide a comprehensive understanding of how the module operates within a real-world data-driven DSS framework.

Intake Phase

After downloading the dataset from the Fluvius website,[4] the user will have a zip file containing a CSV ($READING_2016.CSV$) with the measurement data and a XLSX file ($1_04werkelijkeverbruiksprofielenhuishoudelijkeklanten-elektriciteit2016Legende.xlsx$) containing the metadata. The presented dataset is in Dutch. The first step is to upload the data and the metadata into the

[3] https://www.flandersairesearch.be/en

[4] https://opendata.fluvius.be/explore/dataset/1_04-werkelijke-verbruiksprofielen-huishoudelijke-klanten-elektriciteit/information/

DSS. The intake module receives the CSV, a small extract of it is presented in Figure 2.2, and then the user needs to deliver the metadata. To do so, the system provides a template to fill in. The user is not obliged to complete all the fields to proceed with the analysis, however, the more information provided, the higher the final score.

Data and Metadata Uploaders The template for the metadata designed for the POC has two parts: General information and Data information. General information is for the future usage of the submitted data. It consists of the following:

- **the project's name**: Low Voltage Grid - Forecasting Energy Consumption
- **the domain**: Industrial
- **the problem statement**: The visibility on the low voltage (LV) distribution grids throughout Europe is limited: the layout of the grids is only partially known, and measurements are limited. In the past, this was acceptable, as we employed a "fit and forget" strategy: install (over-dimensioned) cables with sufficient capacity to cover all demand peaks. Today, this solution is not an option. Now, the alternative is to develop technology to use the installed capacity more optimally by operating our grids closer to their limit and technically supporting measures to mitigate the impact of the energy transition.
- **the research goal**: To run long-term forecasts algorithms for all potential evolutions in the grid use [9], [93].

Data information is more related to how the data was retrieved. In the example template, the requested information is:

- **the data type**: Time Series
- **the source**: Fluvius Open dataset
- **the data description**: The dataset contains 100 time series with quarter-hourly offtake and injection measurements of Low-Voltage (LV) grid connections. These anonymized measurements were obtained in a digital meter proof-of-concept project in 2016, which was carried out in a pilot area of the Belgian territory. This dataset only includes the digital meters for which more than 98% of their readings were validated. Given that the total amount of expected measurements is equal to 96 (readings/day) x

366 (days/year) = 35136 (readings/year), the selected digital meters in this dataset have more than 35136 x 0.98 =34433 validated readings.

Then, there was a drop-off space where the data file had to be added. The last step of this phase is to move the CSV file (Figure 2.2) with the data to the proper directory and to write the metadata in a JSON file (Listing 2.1). Then, the system starts the next phase- the assessment phase.

Listing 2.1. **Extract of the JSON metadata file.** The file stores the metadata related to the uploaded dataset. It includes general project information, such as the project name and problem statement, as well as specific data information, including the data type (Time Series) and a description of the dataset used in the proof-of-concept.

```
{"general": {
      "name": "Low Voltage Grid",
      "problem_statemnt": "..."},
  "data": {
      "type": "Time Series",
      "description": "..."}}
```

Assessment Phase

The assessment phase accesses the CSV and the JSON files, the outputs of the previous phase (Section 2.5.2). Then, the three processes of this phase begin. The system automatically executes this phase following the configuration setup by the user.

EU Guidelines Assessment Process The EU guidelines assessment process examines the inputs, comparing them with rules extracted from the European Guidelines [22]. The four dimensions considered in this step are:

- Interoperability: The DSS checks the presence of an encoding issue inside the data and metadata. In the POC, the Python library cChardet [5] was used for this task. The system did not find any encoding issue; the results are illustrated in Table 2.1. The outcome was stored in the JSON result file to be later translated into the final report.
- Findability: The DSS prepares the working space for the Intake Phase's output. The system checks if the project already exists. If so, the metadata file is moved to the already existing project's working space as

[5] https://pypi.org/project/cchardet/

a subproject. If not, a new working space is created. In the POC, the working space was a file system directory composed of a general directory called "Working Space". It contained the project directory, which in turn contained the directories for the data, the metadata, and the future result. From the metadata file, the data's description was examined using text quality indicators implemented in Python [50]. Specifically, the POC implemented the number of Spelling Errors, [6] Lexical Diversity, [7] number of Ungrammatical Sentences, [8] and Average Sentence Length. All the results on the metadata were saved in the JSON result file. The results obtained by the POC are shown in Table 2.2. They give an indication of the clarity of the metadata input. In this case, for example, the lexical diversity, in which the maximum is one, is quite low. This means that there are many repetitions of the same words. This analysis is important for the reusability of the dataset by other users.

- Accessibility: At this point, the DSS has moved the data file to the working space. During this step, as described in Section 2.4, the system checks for any possible security constraints, such as limited clearance access for certain personnel. In the POC, there were no introduced constraints.

- Reusability: The last step is to verify the data itself. In the POC, the system read the content of the CSV file (Figure 2.2) without any problems. The content was loaded as an object called a dataframe.[9] If this operation had caused an error, the process would have read how to mitigate the situation from its associate database. For example, a mitigation action could be asked for the user's involvement. Having the dataframe, the DSS checked the number of samples, the presence of headers, and any other constraints the user might have required. The success or failure of each instruction was saved in the JSON result file.

CSV containing Data	JSON containing Metadata
'encoding': 'ASCII'	'encoding': 'ASCII'
'confidence': 1.0	'confidence': 1.0

Table 2.1. **Results of the Interoperability Check.** ThisTable presents the outcome of the interoperability check performed on the data and metadata during the EU guidelines assessment process. The check, conducted using the cChardet library, ensures that no encoding issues are present in the files. In this case, the system found no encoding issues, confirming the integrity of the data.

[6] https://pypi.org/project/pyenchant/
[7] https://github.com/kieferca/quality-indicators-for-text
[8] https://pypi.org/project/language-tool-python/
[9] `https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html`

Rule	Result
# Spelling Error	4
Lexical Diversity	~ 0.091
# Ungrammatical Sentences	2
Avg Sentence Length	23.5

Table 2.2. **Metadata Quality Indicators.** This table shows the results of the text quality analysis applied to the metadata description during the EU guidelines assessment process. Indicators such as Spelling Errors, Lexical Diversity, Ungrammatical Sentences, and Average Sentence Length were calculated. The analysis provides insights into the clarity and reusability of the metadata, with a low lexical diversity indicating potential redundancy in word usage.

An extract of the JSON file result, including the output of the accessibility and reusability, is shown in the Listing 2.2. To summarize the EU guidelines assessment process, the inputs are the metadata file (JSON file), the data file (CSV file), and the output is the JSON result file, containing the results of each subprocess.

Data Quality Assessment Process During the second process, the DSS examines the quality of the data contained in the CSV input in depth. The analyst input $w_{comp} = 0.3$, $w_{acc} = 0$, $w_{time} = 0.3$ and $w_{cons} = 0.4$ into the DSS. Finally, the threshold ϵ was set to 0.8.

The DQ_i indicators were calculated as follows.

- Completeness: In Section 2.4, this metric (DQ_{comp}) evaluates how much information the dataset has. So, in the POC, it was calculated as

$$\frac{\#\ Expected\ values\ -\ \#\ Missing\ values}{\#\ Expected\ values}. \tag{2.4}$$

  Numerically, it was $DQ_{comp} = \frac{35136-20078}{35136} = 0,43$.

- Accuracy: In the POC, this metric (DQ_{acc}) was not calculated. Accuracy measures the distance between the input data and a reference dataset. Having a reference dataset is not always possible because, among other reasons, it requires a lot of time for the expert to generate a dataset containing what is desirable. Moreover, the representativeness of a reference dataset loses its meaning fast. Energy consumption data depends on many factors; among them, there are human habits and the employment of new technology, such as heat pumps and electric vehicles. The reference dataset should reflect these continuous changes, causing frequent updates that are not sustainable. Therefore, the system, not finding the reference dataset, skipped this evaluation.

- Timeliness: In Section 2.4, this metric (DQ_{time}) was defined as a measurement to evaluate how up-to-date the input is. In the POC, the data available was from 2016 because it is hard to have values from a recent period. Recent studies, [37], underlined that the electricity consumption by households in Belgium had decreased by 6.7% compared to the usage in 2010. On average, there was a decrease of 1.2% in the European Union. Since the data available was from 2016, the analysts decided to consider the data out-of-date because the percentages are significant, however still relevant for the considered use case. Hence, the analyst assigned zero as the timeliness score.

- Consistency: Having the data saved in files is not the most efficient way to work. So, this metric (DQ_{cons}) measures if the user prefers to save the data in an external database and calculates how many semantic rules are violated by the dataset. Figure 2.5 shows the interaction between the DQM framework's architecture and an external database. The depicted architecture is a simplification of the one shown in Figure 2.1. The DQ assessment process (Fig. 2.5C.2) is possible to link to an external centralized database (Fig. 2.5EXT) defined by the user and containing the data. In the POC, at this point, the DSS established a connection with the external relational database. It retrieved the appropriate table definition and inspected each attribute, comparing the type with the pertinent column in the dataset. If the column were a general object, the system would try to convert the values to the expected type. In Table 2.3, two example results of this operation are shown. The Table contains the names of the column in the dataset (Fig. 2.2), which are the same inside the external database as attributes, the types of the attribute found in the external database table's definition, the types of the column in the dataset and the result: they match or they do not. Consistency was measured as follows:

$$DQ_{cons} = \frac{\#\ Checked\ attributes - \#\ Failed\ matches}{\#\ Checked\ attributes}. \tag{2.5}$$

  In numbers, it is $DQ_{cons} = \frac{7-1}{7} = 0,86$.

The result obtained by this process for each dimension was saved in the JSON result file. The calculations of $\overline{DQ}$ (equation 2.3), therefore:

$$\overline{DQ} = 0.3 \times 0.43 + 0 \times 0 + 0.3 \times 0 + 0.4 \times 0.86 = 0.473 \tag{2.6}$$

As $\overline{DQ} \leq 0.8$ the DSS defines the quality level as insufficient.

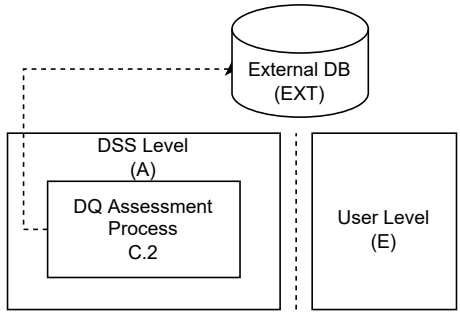


Figure 2.5. **Interaction Between DSS Module and External Database.** This figure depicts how the Data Quality (DQ) Assessment Process interacts with an external relational database (EXT). It shows the data exchange flow and integration between the DSS and the external database, which is used to verify the consistency of the dataset against predefined schema definitions. The diagram is a simplified version of the overall DSS architecture, focusing on the data validation process.

name	attribute:type	column:type	result
InstallatieID	VARCHAR	String	Success
Meetwaarde	FLOAT	String	Fail

Table 2.3. **Consistency Check Results.** This table presents the results of comparing the dataset columns with the expected types defined in the external relational database. The system checks the compatibility between each dataset column and the corresponding attribute in the external database. The Table lists column names, the attribute types found in the external database, the types identified in the dataset, and the result of the match (Success or Fail).

Data Cleaning Assessment Process The DC Assessment Process is the last step before having the final report. During this stage, the DSS inspects some primary data-cleaning issues. The ones implemented in the POC, and already introduced in Section 2.4, are the following:

- Time Column: The DSS analyzes the column containing the timestamps. In the POC, the first action was to find the range of the considered period, so the start and the end date. Then, reading the hours, the DSS identified the time frequency. Additionally, the system examined the time stamps around March and October with particular attention. At the end of these months, in Europe, there is a change of hours between winter and summer time or vice-versa. This time change of one hour could create duplicated or missing timestamps. The DSS verified the presence of this problem. All these findings were stored in the JSON result file to be double-checked with the user.

missing type	result
null values	0
missing rows	20078

Table 2.4. **Missing Values and Rows.** This table displays the results of the system's analysis of missing values and rows in the dataset. The first column shows the types of missing data checked, while the second column indicates the number of occurrences of each type. The analysis includes both null values and missing rows based on the provided timestamps and metadata.

- Single-value columns: In the POC, the DSS, iterating on the columns, identified the column with single values. An example is the column "Type" in Figure 2.2. The value was stored in the JSON Result File, and during the last phase, the system will propose to drop the columns and save the value as metadata.
- Type of columns: This analysis is linked to the CONSISTENCY dimension tested in the previous process. The Consistency test can be performed only if the user provides semantic rules. The DSS automatically tries to recognize the data type if this does not happen during the DC Assessment Process. In the POC, one value of each column was tested to verify if it was a string, boolean, category, int, float, or DateTime.
- Duplicates: In the POC, the DSS calculated the number of duplicated rows found in the system. The user decides how to handle them during the analysis phase, in which all these problems are presented. In the dataframe, the system did not find any duplicated values.
- Missing Values: In the POC, the last examination done by DSS was to verify the presence of missing values. First, the system verified the presence of cells with null values. Then, the system calculated the missing rows using the timestamps column and the information obtained from the metadata. The results are shown in Table 2.4.

An indicator is calculated at the end of this process, similar to the DQ assessment process (Section 2.5.2). This indicator was computed using Equation 2.7. In detail, the system counted the total number of all the cleaning problems checked during the DC assessment process, then it calculated the number of actual cleaning problems that the user needs to check in the next phase. The results are shown in Table 2.5. For example, the test to identify the period range was one, and the system found the answer, so zero problems were needed to be checked in this case. However, during the computation of the missing values, two tests were performed, and one did need human intervention. The final

indicator was computed like

$$DC = \frac{\#\ Checked\ problems\ -\ \#Actual\ problems}{\#\ Checked\ problems}. \tag{2.7}$$

Then, the obtained number was converted to a percentage. A high score indicator means fewer problems to check during the next phase. In the presented scenario, $DC\ Indicator = \frac{17-7}{17} \times 100 = 37\%$.

	actual problems	checked problems
id period range	0	1
id time frequency	0	1
winter/summer	2	2
single values per each column	3	7
type check	1	3
duplicates	0	1
missing values	1	2

Table 2.5. **Data Cleaning Indicator Calculation.** This table summarizes the results used to compute the Data Cleaning (DC) indicator. It lists the types of cleaning issues identified, the number of actual problems found, and the number of problems checked by the DSS. The DC indicator is computed by comparing the number of checked problems with the actual problems that require further user intervention.

Report Generator The assessment phase concludes with the generation of the report, summarizing all the findings gathered during the three processes. The input of this block is the JSON result file, an extract of the one produced by the POC is depicted in Listing 2.2.

This module's subphase aims to allow the user to review the results and, if needed, download them in a report.

The report should summarize each step executed by the DSS until now and contain all the relevant information related to the project.

In the POC, a template was designed and filled with the obtained results. The DSS started generating the final document filling in the project's name. Then, it added the project's description, the data's description, and an extract of the dataset, similar to Figure 2.2. After this, the DQ and DC indicators were shown as the first result. Following that, the metadata evaluation was introduced as in Table 2.2 during the EU guidelines assessment process. All the other results were organized in distinct sections, each for a particular step in the process similar to the pattern used in Section 2.5.2.

Listing 2.2. **Extract of the JSON result file summarizing the findings of the three assessment processes in the Proof-of-Concept.** This extract illustrates the results from the EU Guidelines Assessment process, including details such as encoding checks, metadata analysis, findability indicators, accessibility visibility, and reusability evaluation. The data sections represent key findings for each process, such as encoding confidence, metadata quality, and accessibility status. This JSON file serves as the foundation for generating the final report.

```
{"EU": {
  "INTEROPERABILITY":
      {"Data":
        {"encoding":"ASCII",
        "confidence":1.0},
      "Metadata": {"..."}},
  "FINDABILITY":
      {"Sub_project":"NO",
       "Location_workspace":"..." ,
       "metadata_indicator": ,
       "metadata_details": {
         "Spelling":"2",
         "..."}},
  "ACCESSIBILITY": {"Visible":"All",
      "Modificable":"All"},
  "REUSABILITY": {"data":"Well Formed"}}
```

Analysis Phase

The last phase in the proposed DSS module framework is the analysis phase. This phase is designed in a semi-automatic fashion. The user is involved in every step and actively participates in the decision-making process. The system supports the user in gathering information and underlining potential decisions to make. The input is the JSON result file acquired from the assessment phase; the output is a series of preprocessing steps designed to improve the quality of the data. Figure 2.1D shows the processes composing this phase.

The issue extractor has the job of formulating the possible cleaning problems based on the JSON result file and based on the experiences collected inside the reproducibility database. In the POC, this database was designed as a relational database. It was used to store all the instructions executed by the user to investigate and solve a cleaning problem. The information stored inside the database is the following: the "id", which represented the identification of the historical record, the "datatype", which was the type of data used for that specific record, the "project", which was the name of the analyzed project,

the "problem", which was the name of the data cleaning issue the DSS and the user found, the "solution", which was the code performed by the user to solve the cleaning problem, and the "info", which was the code used by the user to visualize the data and investigate the problem. All the attributes are strings, except "id" which is an integer, and "solution" and "info" which are JSON. The issue extractor took from the JSON result file the data type, Time Series, and the first problem to check with the user, for example, "Missing Values." Then, the process searched for entries in the reproducibility database with datatype=Time Series and problem=Missing Values; if there were any past experiences, all the information was passed to the analyzer.

The analyzer is the interface between the system and the user. In the POC, the GUI was divided into four spaces:

1. Visualization space was where the DSS results of the under-examination problem were shown. The DSS showed the Missing Values test results during the DC assessment process, Section 2.5.2.

2. Experiences space was where the DSS showed the experiences found in the reproducibility database, if any. Firstly, it offered the possibility of visualizing the analysis done. Therefore, the system proposed the solutions.

3. Study space was where the user performed instructions to investigate the problem. The analyst could start from a code proposed in the previous space or propose a personal investigation.

4. Solution space was where the user could accept a past solution or write a new one. For example, the proposed solution could handle the missing data using the interpolation technique. The user could refuse it and instead use the personal solution of filling in the missing values by copying the preceding values.

The interface shown in Figure 2.6 represents the intended user interaction with the DSS module. While an initial version of the analyzer component has been implemented, the figure presents a conceptual mock-up that reflects the envisioned structure and functionalities of the final user interface. It highlights how the system guides the analyst through automated suggestions, exploratory tools, and historical insights in a cohesive environment.

When the user was satisfied with the solution, all the new feedback was passed to the issue reformulator. This process had the job of creating the entry for the reproducibility database. It translated the Python code written in the study space and the solution space into two JSON objects. Then, it gathered all

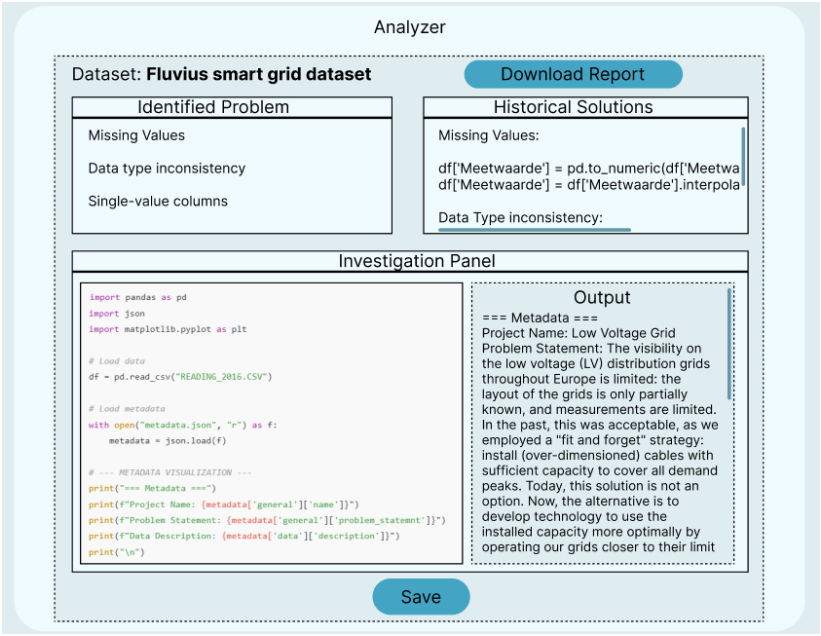


Figure 2.6. **Example of the Data Cleaning (DC) investigation interface.** This illustrative mock-up presents the conceptual UI of the data quality improvement module developed for the Proof-of-Concept (POC). It includes visual summaries of identified data issues (e.g., missing values, type inconsistencies), proposed historical solutions, and an interactive panel for reviewing metadata and conducting exploratory analyses. The interface demonstrates how the DSS supports the analyst by combining visual feedback and code-based investigation.

the information to complete the input query and sent it to the database. To conclude the cycle, the issue reformulator process notified the issue extractor of the success of the operation, and so a new problem is passed to the analyzer.

After showing all the problems examined by the DSS during the assessment phase, the system checks the presence of different problems related to the datatype within the reproducibility database. If any, these are shown to the user. This step helps review problems that, maybe, the analyst has not thought about. In the end, the user could propose new problems to test, and they would be saved as new experiences in the database.

2.5.3 Implementation Details

- **Technology Stack** For the prototype implementation, the following technologies were used.
    - o **Frontend**: Streamlit for the graphical user interface (GUI).
    - o **Backend**: Python for processing logic.
    - o **Database**: JSON files for metadata storage, relational databases for reproducibility records.
    - o **Main Libraries**: Pandas for data manipulation, cChardet for encoding detection.
- **Test Cases Performed** To check the internal consistency and illustrate its feasibility, the system was tested using the following use cases:
    - – **Data Upload**: Verification of different data formats (CSV, JSON) and file sizes to ensure correct ingestion.
    - – **Data Quality Evaluation**: Ensuring the correctness of quality assessments against known datasets.
    - – **Cleaning Assessment**: Checking the identification of common data cleaning issues, such as missing values and duplicates.
    - – **Report Generation**: Ensuring the generated reports contain all necessary details and are correctly formatted.
    - – **Usability and User Experience**:
        - ∗ **User satisfaction and task completion efficiency testing was considered but not performed, as the prototype was developed for research purposes only and not for direct end-user deployment.**
- **Limitations and Constraints**
    - o **Incomplete Features**
        - – Some advanced data quality metrics are not yet implemented.
        - – Real-time data processing is currently not supported.
    - o **Known Issues**
        - – Limited integration with external databases for some data types.
        - – The user interface may require optimization for handling large datasets.

2.6 Conclusions

Data Quality (DQ) and Data Cleaning (DC) are fundamental for professionals working with data, particularly in Artificial Intelligence (AI) and Decision Support Systems (DSS). This chapter introduced the DQM framework, a structured approach that assists analysts in evaluating, understanding, and improving data quality while optimizing preprocessing workflows. As described in Section 2.4 and applied in Subsection 2.5.2, the computation of key indicators for DQ and DC enabled the framework to provide users with valuable insights, ensuring better resource allocation during the data-cleaning process.

One of the main objectives of this research was to minimize redundant cleaning workflows by promoting reproducibility. The DQM framework addresses the common issue of developing custom solutions from scratch, which leads to inefficiencies and an increase in cost (Haug et al. (2011)) [44]. By integrating a historical repository of cleaning transformations and user feedback, the DQM framework enables analysts to reuse prior solutions and, in this way, optimize workflows and significantly reduce preprocessing time, as described in Subsection 2.5.2.

Furthermore, in Section 2.5.2, it was shown how the DQM framework improves the integration of human experience with automation. Most existing systems either rely entirely on predefined rules or provide only limited adaptability to different datasets. By incorporating a human-in-the-loop approach, the framework bridges the gap between automated assessments and expert decision-making, enhancing both flexibility and precision in handling data issues (Section 2.5.2). This interactive feedback loop ensures that expert knowledge is continuously captured and leveraged for future datasets, improving long-term efficiency. As Rezig et al. (2019) [77] emphasized in their work, a human-centric approach is fundamental in algorithm-agnostic data cleaning frameworks because most current systems tightly couple human involvement with specific cleaning algorithms, limiting their ability to generalize across different data cleaning tasks. By contrast, a human-in-the-loop framework should be designed to systematically integrate expert knowledge in a way that is independent of the underlying cleaning techniques, allowing greater adaptability to new datasets and error types. The DQM framework has been shown to achieve this by interacting with the user in the analysis phases where the system suggests highly probable problems relevant to the dataset being processed, as discussed in Section 2.5.2.

Another key contribution of the framework is its enhanced reporting mechanisms, which summarize DQ assessments and suggested preprocessing strategies, as presented in Subsection 2.5.2. Analysts receive clear, structured feedback

on data issues, making it easier to assess data readiness and design effective cleaning strategies. This improvement directly addresses the lack of clarity and standardization in many existing DQ tools, facilitating better decision-making in DSS environments. As Bammidi et al. (2024) [5] stated, data quality plays a crucial role in decision-making systems, as it directly impacts the reliability of insights derived from data. A correct and indicative evaluation of data quality allows analysts to choose the right dataset for the right application, ensuring that decision-support models are built on trustworthy and well-prepared data. The DQM framework aligns with this principle by providing detailed, interpretable reports that assist analysts in understanding data quality deficiencies, recommending preprocessing steps, and ultimately enhancing the overall decision-making process, as discussed in Section 2.5.2. From a theoretical point of view, the framework has proved to be appropriate and efficient to be adapted and run to perform the tasks and meet the objectives set in the specific context mentioned above. Therefore, it could be claimed that the case study demonstrated the system's practical utility and adaptability. These qualities reinforce the DQM framework's potential impact on real-world AI and DSS applications.

The framework was validated through a Proof-Of-Concept (POC) application in an industrial setting, focusing on low-voltage grid management using public time series data from Belgium's Fluvius dataset. While certain metrics, such as accuracy, were not always directly applicable, the framework remained valuable in structuring the data-cleaning process and improving data usability. The case study demonstrated the system's practical utility and adaptability, reinforcing its potential impact on real-world AI and DSS applications.

Future work will focus on broadening the applicability of the framework, improving automation and learning capabilities, and integrating it into a larger decision support ecosystem.

One of the primary areas of future research is ensuring the generalization of the framework across multiple datasets and domains. The current validation was performed in the context of low-voltage grid management using the Fluvius dataset, but expanding its applicability to other fields will be crucial in assessing its robustness. For example, applying the framework to healthcare datasets could test its ability to handle missing values, inconsistencies, and data heterogeneity, as commonly seen in electronic health records and multimodal medical data.

Beyond generalization, an important direction for future work involves enhancing the learning mechanisms of the framework to improve automation and decision support. The historical repository of transformations and user feedback already provides a foundation for reusability, but additional efforts will be made to integrate machine learning techniques to predict optimal cleaning strategies

based on past data preparation efforts.

Another crucial aspect to address in future work is the development of more advanced user interaction mechanisms. The human-in-the-loop approach already improves the integration of expert knowledge with automation, but refining how users interact with the framework will make it more effective. One possible improvement is the implementation of interactive dashboards that allow users to visualize data quality issues in real time and adjust preprocessing parameters dynamically. Additionally, developing collaborative features that enable multiple analysts to share and compare data cleaning processes could facilitate standardization and reproducibility across teams working with similar datasets.

While the DQM framework currently assists in preprocessing and DQ assessment, its impact can be significantly amplified if embedded into a broader decision-making pipeline, as shown in Chapter 4.

Chapter 3

DSS4EX: Framework to explore Artificial Intelligence Pipelines

This chapter is a published paper which has been modified and reprinted with permission. The original version can be found on:

Rinaldi, Giulia; Theodorakos Konstantinos; Crema Garcia, Fernando; Agudelo, Oscar Mauricio; De Moor, Bart
"DSS4EX: A Decision Support System Framework to Explore Artificial Intelligence Pipelines with an application in Time Series Forecasting".
Expert Systems with Applications, ISSN: 0957-4174, 2025
DOI: 10.1016/j.eswa.2025.126421
https://www.sciencedirect.com/science/article/pii/S0957417425000430

Abstract

Understanding complex Artificial Intelligence (AI) pipelines for time series forecasting can be challenging for both experts and non-experts. This work introduces DSS4EX, a Decision Support System (DSS) framework designed to facilitate the exploration and comprehension of AI pipelines. The framework is demonstrated through a software demo currently in a prototype phase, specifically tailored for electricity demand forecasting that utilizes the Decomposition-Residuals Deep Neural Network (DR-DNN) pipeline. The dataset used in the software demo, spanning March 18th, 2017, to February 16th, 2021, contains seven hourly time series from an unknown location. It covers 34,360 timesteps of data, including power demand, air pressure, cloud coverage, humidity, temperature, wind direction, and wind speed. The prototype software demo was created to showcase the usability of the DSS4EX framework, allowing users to interactively configure, visualize, and understand the AI pipeline. This interactive approach enhances user engagement and comprehension, promoting informed decision-making. The DSS4EX framework makes complex models more transparent and accessible, bridging the gap between advanced AI models and user understanding. This research demonstrates the potential of DSS4EX in supporting further advancements in AI applications by making sophisticated AI techniques more understandable and usable for a broader audience.

3.1 Introduction

Even though the term Decision Support System (DSS) has been used since the 1980s [95], in the last 20 years, these types of software have increasingly been applied to design pipelines for time series forecasting, a process that predicts future values based on past data trends. Time series forecasting is widely used in fields like medicine and industry to anticipate future outcomes and make data-informed decisions. In the medical field, DSS has helped improve patient care for conditions like heart failure [7] and supports personalized treatment, prediction, and information gathering [49]. In the industrial sector, DSS frameworks help convert knowledge into new models, often by building on existing ones [43]. For example, data mining tools have been integrated into DSS for farm management [81]. These systems use Artificial Intelligence (AI) to help users understand situations, choose appropriate methods, and use processed data for results.

Traditionally, DSS has used AI to solve specific problems, with human users leveraging AI outputs to improve decisions. Recently, interest has grown in understanding AI itself, especially how models produce certain results. This interest has extended to AI experts and others, such as researchers in Explainable AI (XAI) and Human-Computer Interaction (HCI), who are working to make AI systems more transparent and trustworthy [65]. As a result, DSS tools originally built for decision support can now be adapted to help users explore and better understand complex AI processes.

This chapter proposes the design of a new framework called DSS4EX, a Decision Support System (DSS) for exploring AI pipelines. The key contributions of this work are outlined as follows:

- **Tailored exploration of AI pipelines**: DSS4EX is specifically designed to explore and understand complex AI pipelines. Unlike traditional DSS, which primarily focuses on aiding decision-making, DSS4EX emphasizes the exploration and comprehension of AI methods, making it unique in its approach to AI transparency and user interaction.
- **Enhanced information flow transparency and user understanding**: It is essential when working with complex AI systems for a better reproducibility and understanding of how information is processed.
- **User accessibility with expert guidance**: DSS4EX is accessible to users of all levels, from novices to experts, through an intuitive graphical user interface (GUI). This inclusivity is further supported by the integration of expert guidance within the system.

- **Practical application in time series forecasting**: The practical application of DSS4EX is demonstrated through a software demo tailored for electricity demand forecasting using the Decomposition-Residuals Deep Neural Network (DR-DNN) pipeline. This real-world example shows the framework's usability and effectiveness in handling complex time series forecasting problems.

In conclusion, DSS4EX's interactive design allows users to configure, visualize, and understand AI pipelines, improving user engagement and comprehension. This interactivity helps make sophisticated AI techniques more understandable and usable for a broader audience by bridging the gap between advanced AI models and user understanding. This chapter focuses on the DSS architecture and its contributions, with the software demo as an example of functionalities that could be expanded to include different Neural Network architectures for comparison.

It is important to emphasize that DSS4EX is currently in a prototype phase and not a fully developed software for large-scale applications. The primary purpose of the DSS4EX demo is to serve as a proof-of-concept, demonstrating how such a system could support non-technical users in understanding, exploring, adapting and improving AI pipelines. At this stage, DSS4EX is being tested with an energy time-series forecasting scenario to highlight its usability and potential for user engagement. While trustworthy tools that help users understand AI are not a new idea, DSS4EX takes a unique approach by focusing on accessibility for non-technical users, setting it apart from tools aimed at professional use.

In contrast to established tools like RapidMiner[1], Knime[2], and IBM SPSS[3], which are commonly used by professionals to build and refine machine learning models, DSS4EX supports non-expert users in gaining insights into the mechanisms and applications of machine learning models. These professional tools streamline AI pipeline development for expert users but do not focus on helping users understand underlying model processes. DSS4EX, however, has a different goal, aiming to make AI comprehensible for a broader audience, particularly those without a technical background.

A unique feature of DSS4EX is its human-in-the-loop approach, which not only informs users about ongoing processes but also provides expert-driven suggestions to aid understanding. In typical systems of this kind, users input information for the system to refine its processes. However, in DSS4EX, this approach also enables the system to offer insights from past expert decisions, suggesting improvements or adjustments to enhance the configuration process.

[1] `https://altair.com/altair-rapidminer`
[2] `https://www.knime.com/`
[3] `https://www.ibm.com/spss`

While other software may rely on documentation, DSS4EX delivers concise, context-specific suggestions directly within the Graphical User Interface (GUI), creating a more intuitive experience. This approach empowers non-expert users by reducing information overload and technical jargon, enabling them to understand processes and make informed decisions. Expert users may also find the responsive, relevant prompts more helpful than extensive manuals.

DSS4EX, therefore, allows researchers to share their work in a way that is accessible to non-expert users, enhancing their understanding of complex AI topics and fostering broader engagement with current technological advancements.

This chapter is organized as follows. Section 3.2 discusses state-of-the-art tools for AI pipeline training, AI concepts, and DSS frameworks. Section 3.3 outlines the research objectives and challenges. Section 3.4 presents the DSS4EX framework. Section 3.5 illustrates and discusses DSS4EX's application as a Web-based Decision Support System (Web-DSS) with the Decomposition-Residuals Deep Neural Network (DR-DNN) pipeline for electricity consumption forecasting [102]. Section 3.6 discusses the results and findings. Finally, Section 3.7 provides concluding remarks and future research directions.

3.2 Literature Review

3.2.1 Software Tools for AI Pipeline Training

The development and training of AI pipelines are critical processes that require robust software tools for effective deployment and monitoring. An AI pipeline is a modular end-to-end architecture that integrates AI techniques to solve specific problems.

Automated Machine Learning (AutoML) aims to automate the training and design of ML pipelines, enhancing productivity and making ML accessible to non-experts [84]. Tools like PyCaret help automate data preprocessing, model selection, and hyperparameter tuning, enabling non-experts to use complex techniques without extensive technical knowledge [112]. Studies on AutoML tools like AutoGluon [87], Auto-Sklearn [40], and PyCaret [4] have shown varied results across different datasets, underscoring the importance of dataset-specific considerations [108]. While these tools aim to democratize AI, they still require some expertise in machine learning. Unlike these AutoML frameworks, which are primarily code-based, DSS4EX was developed with a graphical user interface

[4] `https://pycaret.org/`

to enhance user-friendliness and accessibility. This visual interface design allows users to engage with the tool more intuitively without needing coding knowledge. Moreover, while existing AutoML tools offer some intuitive features, they do not typically provide comprehensive, fully graphical interfaces that support configuration, exploration, and customization of ML pipelines at a non-technical level. This distinction enables DSS4EX to target a broader audience, including users with no prior ML experience, expanding its usability beyond traditional AutoML audiences.

Platforms like DeepTSF and FASTMAN-JMP also aim to simplify pipeline training. DeepTSF uses a codeless approach for time series forecasting, integrating various ML techniques, including Long Short-Term Memory (LSTM) networks and SHapley Additive exPlanations (SHAP) [5], to enhance model transparency [70]. FASTMAN-JMP supports data exploration and fault detection with tools for data cleaning, normalization, and clustering, making it accessible to users without deep ML knowledge [101]. Unlike these tools, DSS4EX emphasizes guided decision support, enabling users to understand the impact of specific parameters through integrated expert feedback, which adds depth to the interpretability of model choices.

The DSS4EX framework proposed in our research shares similarities with these tools but stands out in several key aspects. DSS4EX targets a broader audience, including non-technical users, and incorporates expert guidance to help users understand parameter significance. It also focuses on clear communication protocols and information exchange among system components. The framework is designed for specific problem-solving, demonstrated by a software demo using the Decomposition-Residuals Deep Neural Network (DR-DNN) for electricity demand forecasting. This demo illustrates how users can configure, visualize, and comprehend AI pipelines effectively. DSS4EX is designed with a user-friendly interface and provides expert tips, helping users set up and understand complex pipelines without needing coding skills.

In summary, DSS4EX offers a unique combination of accessibility, expert guidance, and comprehensive framework design, making it particularly suited for a wider audience.

3.2.2 Software Tools for Public Understanding of AI

Artificial Intelligence (AI) is increasingly integrated into various aspects of daily life, including banking, transportation, education, electronics, and healthcare [14]. As AI becomes more common, tools that help the general public understand

[5] `https://shap.readthedocs.io/`

and use these technologies are increasingly needed. Large Language Models (LLMs), such as GPT-3, have significantly contributed by providing advanced capabilities for natural language understanding and generation [23]. Unlike general AI models like LLMs, which focus on natural language processing, DSS4EX specifically addresses AI interpretability in decision support systems, offering an accessible pathway for users to understand and interact with complex and advanced AI processes.

Despite these advancements, the public often views AI models as a "black box" and remains skeptical [15, 11]. Even experts, such as medical doctors, use AI with caution due to its opacity [6, 68, 66]. Understanding AI can facilitate the adoption of these AI-augmented systems [58]. Educational software tools play a critical role, although they often target academic settings and are not problem-driven [96]. Tools like AISpace2 and CIspace help in understanding general AI concepts but are limited in their application scope [115, 3]. In contrast, DSS4EX is specifically designed to support problem-driven exploration, enabling users to apply AI within targeted, real-world energy data rather than in theoretical or academic-only contexts.

Explainable AI (XAI) aims to make AI systems transparent and understandable, helping stakeholders gain insights into the decision-making processes [47]. XAI is essential for creating trustworthy AI systems. In time series forecasting, tools like those described by Maarif et al. (2023) [59] combine LSTM neural networks with XAI techniques to forecast energy consumption and provide interpretability using SHapley Additive exPlanations (SHAP). Zhang et al. (2024) [114] developed ShapTime, an XAI tool for time series forecasting that uses Shapley values to explain the importance of different time periods. DSS4EX builds on these approaches by integrating SHAP with human-in-the-loop guidance, enabling non-technical users to understand model behavior while configuring and adapting the pipeline through both visual explanations and expert-driven insights.

Other XAI systems focus on practical applications. Wang et al. (2023) [107] described an XAI system for hospital recommendations that enhances transparency through various visualization techniques. Clement et al. (2023) [21] presented XAIR, a framework for integrating XAI methods into the software development process, aiding in requirement analysis, design, implementation, evaluation, and deployment. These systems primarily target researchers and practitioners, categorizing and evaluating different XAI tools to improve understanding and usability in real-world applications. Unlike these frameworks, which are geared towards professional users, DSS4EX provides an intuitive interface combined with natural language guidance, broadening accessibility to non-technical audiences and facilitating a deeper understanding of AI decision-making.

This chapter proposes a framework that aims to guide users in comprehending complex AI pipelines, not only for experts but also for a wider audience. The software demo includes XAI techniques, such as Shapley, and provides guidance on parameter significance using natural language, making AI more accessible to the general public. DSS4EX is distinct in its comprehensive approach, combining an intuitive graphical interface with expert-driven insights designed specifically for broader accessibility. Unlike typical XAI tools, which are often targeted at professionals with prior knowledge, DSS4EX offers human-in-the-loop guidance and natural language explanations to make AI pipelines understandable for users without technical backgrounds.

3.2.3 Decision Support System Framework

Decision Support Systems (DSS) are computer-based tools that aid decision-making by providing relevant data, analytical tools, and models. DSS frameworks support a range of decisions from structured to unstructured problems. Integrating AI into DSS enhances their capabilities, allowing for advanced analysis and predictive modeling [36, 39].

A robust DSS framework includes several key components: a Database Management System (DBMS) to manage data storage and retrieval, a Model Management System (MMS) to handle analytical models, a Knowledge Management System (KMS) to combine information from various sources to generate insights, and a User Interface (UI) to facilitate user interaction with the system [105]. While conventional DSS structures rely on these core components, DSS4EX introduces an additional layer of guided insights through human-in-the-loop feedback, enhancing user comprehension and enabling users to make data-informed decisions without prior expertise.

Recent studies have focused on the design principles of DSS. Sjöström et al. (2024) [90] described a Clinical Decision Support System (CDSS) for predicting heart failure readmissions using machine learning. They introduced design principles such as contextual integration, actionable insights, and adaptive explanation levels to ensure the system's relevance and usability for healthcare professionals. Dellermann et al. (2019)[25] presented a Hybrid Intelligence Decision Support System (HI-DSS) for startups. This system combines machine intelligence with human feedback to help validate business models, emphasizing iterative validation and dynamic guidance. Elragal et al. (2024) [32] explored integrating data and algorithms into decision-making processes, emphasizing design principles for data-driven decision-making platforms that enhance human-machine collaboration, accountability, and scalability, aiming to improve decision quality with big data and advanced analytics. Zielinska et al. (2024) [116]

highlighted the importance of human-centered design in AI-driven CDSS, focusing on user research, iterative design, and continuous feedback to ensure the systems are transparent, interpretable, and trustworthy. DSS4EX differs from these domain-specific and organization-centric frameworks by focusing on adaptability for a broad audience, providing intuitive, guided insights that help non-experts engage with complex AI models in various contexts.

These studies underscore the importance of human interaction and feedback in DSS design. Unlike these organization-centric approaches, our framework targets non-expert users. We provide general guidance for developing software tailored to specific problems, emphasizing an Explorative Management System that supports knowledge enhancement rather than just user feedback integration. In contrast to traditional DSS designs that prioritize organizational feedback, DSS4EX focuses on empowering individual users by enhancing their understanding of AI processes, offering exploratory tools that improve decision-making without requiring extensive expertise.

3.3 Chapter Objectives and Challenges

- **Problem Statement: Advancements in AI and DSS**
  - Modern software and technology aim to simplify computations and support experienced data science researchers.
  - Despite progress, making AI and complex systems accessible to non-experts remains a significant challenge.
- **Proposed Solution: DSS4EX Framework**
  - DSS4EX is a Decision Support System (DSS) framework designed to help both experts and non-experts explore and understand AI pipelines.
  - The framework simplifies AI workflows and enhances accessibility through guided interactions and clear visualization.
- **Key Challenges Addressed**
  - **Complexity of AI Pipelines**: Non-experts often struggle to understand AI workflows and their decision-making processes.
  - **Communication Between System Components**: Effective information exchange and structured communication protocols are necessary for system usability.
- **Prototype: Software Demo**

  - A software demo was developed to demonstrate DSS4EX in a real-world use case.
  - The selected AI pipeline, **Decomposition-Residuals Deep Neural Network (DR-DNN)**, was created by Theodorakos et al. (2022) [102] for electricity demand forecasting.
  - Users can configure, visualize, and comprehend the AI pipeline with integrated expert guidance.
  - The prototype serves as a practical illustration of the framework's usability rather than a fully developed product for large-scale deployment.

- **Research Objective: Accessibility and Usability**
  - The goal is not only to propose the framework but also to ensure its accessibility to a broader audience.
  - DSS4EX provides an alternative way for researchers to present AI workflows, improving understanding for less-experienced users.
  - The approach fosters wider adoption and comprehension of advanced AI technologies by non-technical users.

3.4 The DSS4EX Framework

This section introduces the design guidelines of the framework for a Decision Support System for EXploring (DSS4EX) Artificial Intelligence (AI) pipelines. Figure 3.1 shows a high-level overview of the framework, illustrating that the DSS receives data as input and produces a customized AI pipeline and results from running this pipeline with the input data. The process includes a feedback loop where the user evaluates the results. If the results do not meet expectations, the user can modify and re-run the pipeline configuration.

Figure 3.2 provides a more detailed view of the interaction between the user and the DSS. The user begins by starting a project and assigning it a unique name through the User Interface (UI) (Fig. 3.2A). The UI serves as the main communication channel between the user and the system, as described in Section 3.2.3. For optimal usability, the UI should be intuitive, with a layout that displays the pipeline as a sequence of blocks. Given the level of user involvement, several screens on the UI are to be displayed to guide the user through the different stages of the DSS structure. For instance, the first screen may allow the user to start a new project and provide details, such as a name and description, while a second screen is dedicated to configuring the pipeline. An example of the UI is shown in Section 3.5.

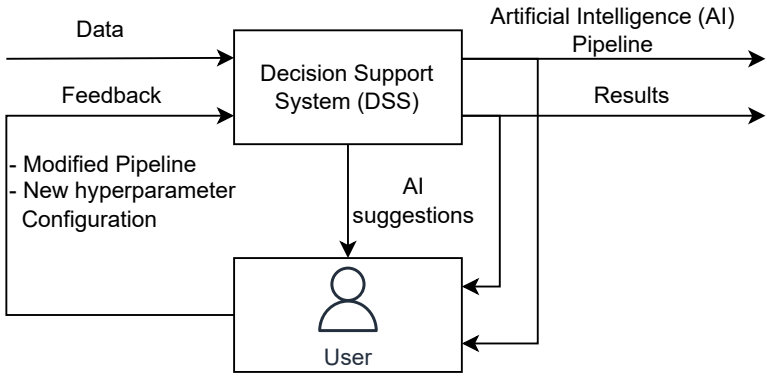


Figure 3.1. **High-Level Overview of the proposed framework. The input to the Decision Support System (DSS) is the data.** The outputs are the Artificial Intelligence (AI) pipeline customized by the user, and the results are obtained by running it with the provided data. These outputs, along with AI suggestions, are presented to the user. The user then decides on the next course of action, which may involve requesting modifications to the pipeline or adjusting hyperparameters for specific blocks.

Since the data is provided by an external source, it is directed into the system through the Data Management subsystem (Fig. 3.2B), bypassing the UI. The Data Management subsystem (Fig. 3.2B) stores and retrieves raw datasets from a database and manages information entered by the user. This subsystem organizes data into three types of databases. The first is the Database (Fig. 3.2H), which contains the datasets. The second is the User's Database (Fig. 3.2I), holding all information related to users and project sessions, such as pipeline configurations and project descriptions. The third is the DSS Database (Fig. 3.2J), which stores important knowledge for the DSS functionalities, including expert insights and pipeline instructions.

The framework has three main information flows. The first flow, called the Intake Flow (Fig. 3.2E), is where the user provides initial details to start a new project. This information is managed by the Data Management subsystem (Fig. 3.2B) and saved in the database (Fig. 3.2H). During this process, the user is the main active participant, inputting information into the system. After completing the Intake Flow, the user proceeds to configure the pipeline, activating the Analysis Flow.

The Analysis Flow (Fig. 3.2G) then triggers the Assessment Flow (Fig. 3.2F). The Analysis Flow manages the user's actions during pipeline configuration, while the Assessment Flow (Fig. 3.2F) delivers suggestions based on expert knowledge and saves the user's feedback. The Analysis Flow (Fig. 3.2G) is linked

to the Model Management subsystem (Fig. 3.2D), which, unlike traditional frameworks described in Section 3.2.3, consists of two parts: the Topological (Fig. 3.2D.1) and Hyperparameter (Fig. 3.2D.2) entities. The Topological entity manages the structure and the relations of the pipeline blocks, while the Hyperparameter entity handles the configurations needed for each block to execute correctly. Some examples of Hyperparameter entities may include lags, hidden size, learning rate, and others. The Model Management subsystem (Fig. 3.2D) is also responsible for saving and running the configured pipeline.

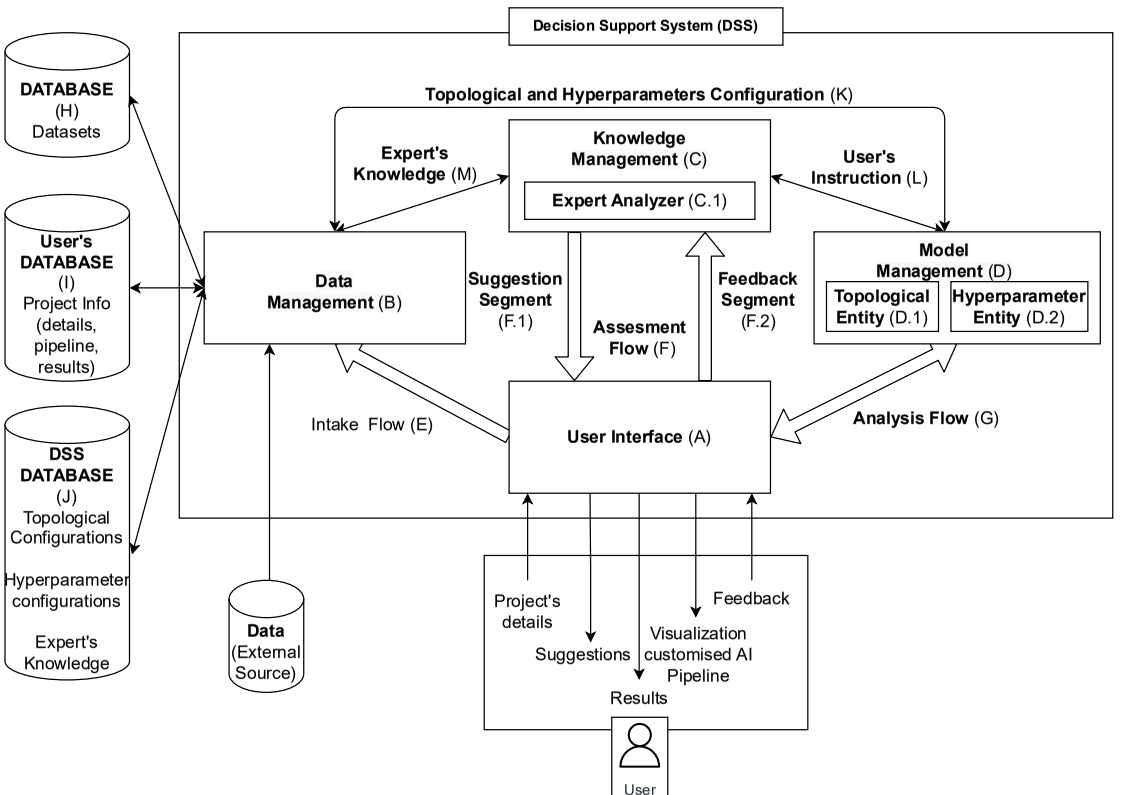
The Assessment Flow (Fig. 3.2F) consists of two segments: the Suggestion (Fig. 3.2F.1) and the Feedback (Fig. 3.2F.2) segments. These two segments function independently and at different stages. The Suggestion segment (Fig. 3.2F.1) activates during the Analysis process, offering guidance when the user selects a pipeline block to configure. The Feedback segment (Fig. 3.2F.2), however, is triggered when the user reviews the final results produced by the Model Management subsystem (Fig. 3.2D). At this point, the user can provide feedback through the Feedback Segment (Fig. 3.2F.2), indicating whether changes to the pipeline configuration are needed. If so, the Analysis process (Fig. 3.2G) resets the pipeline; otherwise, all processes terminate, and the Feedback segment saves the information in the DSS database.

The Assessment Flow (Fig. 3.2F) is directly linked to the Knowledge Management subsystem (Fig. 3.2C), which includes the Expert Analyzer (Fig. 3.2C.1). The Expert Analyzer accesses stored expert knowledge and retrieves relevant information to make suggestions, which are then sent to the UI (Fig. 3.2A) via the Suggestion Flow (Fig. 3.2F.1). Additionally, the Knowledge Management subsystem (Fig. 3.2C) is responsible for saving the latest configuration and results that the user wants to keep.

Each subsystem is connected to others through specific links. The Topological and Hyperparameter Configuration Link (Fig. 3.2K) transfers information about parameter settings and pipeline structure from the Data Management subsystem (Fig. 3.2B) to the Model Management subsystem (Fig. 3.2D). The User's Instruction Link (Fig. 3.2L) conveys user choices from the Model Management subsystem (Fig. 3.2D) to the Knowledge Management subsystem (Fig. 3.2C) for tailored suggestions. During this process, the connection (Fig. 3.2M) between the Data Management subsystem (Fig. 3.2B) and the Knowledge Management subsystem (Fig. 3.2C) retrieves expert knowledge from the database.

Having discussed what the subsystems and the links are and their function, Section 3.5 discusses how to use the framework in a practical application using a deep learning pipeline on time series data.

Figure 3.2. **The DSS4EX framework.** The DSS4EX framework represents a Decision Support System (DSS) tailored for systems that aid users in exploring Artificial Intelligence pipelines. It comprises the classical subsystem of a DSS, but the Knowledge Management subsystem and the Model Management subsystem were reinforced with more details specific to the scope. The interaction between the interface and each subsystem is described by the flow of information interchanged.



3.5 An industrial application of the framework as software demo: DSS4EX for the Decomposition Residuals Deep Neural Network pipeline

The practical applicability of the DSS4EX framework is demonstrated through a real-world industrial use case, focusing on time series forecasting for electricity demand. This software demo integrates the Decomposition Residuals Deep Neural Network (DR-DNN) pipeline, showcasing how users can interact with, configure, and analyze an AI-driven forecasting model through the DSS4EX interface. By leveraging expert insights and guided workflows, the demo allows users to experiment with different configurations, interpret results, and optimize AI pipeline settings without requiring extensive programming expertise.

This section outlines the industrial relevance of DSS4EX by detailing its implementation within the energy sector. The DR-DNN pipeline, originally developed for electricity consumption forecasting, serves as the test case for the DSS4EX prototype, demonstrating how the framework enhances model transparency, reproducibility, and user accessibility. Through this application, the DSS4EX software demo provides a hands-on approach to bridging the gap between AI-driven decision support and industry adoption.

3.5.1 General Description

- **Software Demo Title**: DSS4EX for the Decomposition Residuals Deep Neural Network pipeline
- **Overview**: DSS4EX is a Decision Support System (DSS) designed to facilitate the exploration, configuration, visualization, and comprehension of complex AI pipelines. The prototype focuses on time series forecasting for electricity demand using the Decomposition-Residuals Deep Neural Network (DR-DNN) pipeline. It aims to make AI techniques more understandable and accessible for both experts and non-experts.
- **Objectives**:
    - Provide an intuitive interface for configuring and visualizing AI pipelines.
    - Enhance user understanding of AI methodologies through expert guidance.
    - Demonstrate a practical application using the DR-DNN pipeline.

- Enable comprehensive exploration and comparison of different AI pipeline configurations.

- **Scope**:
  - **Targeted Functionalities**: The system covers the following main functionalities.
    - GUI for configuring, visualizing, and understanding AI pipelines.
    - Integration with expert insights for guided pipeline configuration.
    - Support for time series data analysis and forecasting.
    - Modular interface for pipeline adjustments.
    - Features for downloading models and configurations.
  - **Excluded Functionalities**: The following functionalities are not included in the current software demo because they have not been regarded as essential for showing the framework's effectiveness.
    - Large-scale deployment features.
    - Support for domains beyond time series forecasting.
    - Automated hyperparameter optimization (future work).

- **System's Requirements**
  - **Functional Requirements**: In order to run the framework's processes, the system must provide the following functionalities.
    - Data management capabilities, including input, storage, and retrieval.
    - Configurable AI pipeline components with expert-driven recommendations.
    - Interactive visualization of AI pipeline structures and configurations.
    - Execution and monitoring of AI pipeline training processes.
    - Analysis and visualization of results using SHAP plots.
    - Capability to compare different pipeline configurations.
  - **Non-Functional Requirements**: The system, additionally, must meet the following non-functional requirements.
    - **Performance**: Optimized performance for loading interfaces and processing configurations.
    - **Security**: Secure storage and controlled access to user data.
    - **Usability**: Intuitive and user-friendly UI design.
    - **Scalability**: Scalability to handle various dataset sizes and complex configurations.

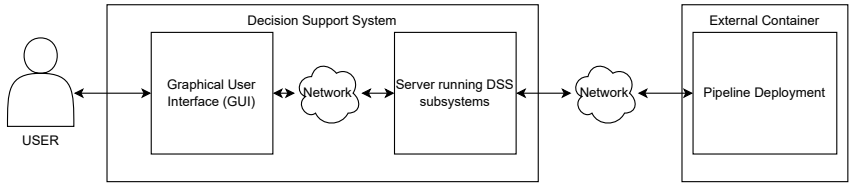


Figure 3.3. **Architecture of DSS4EX.** This figure illustrates the client-server architecture of DSS4EX, where the user interacts with the Decision Support System (DSS) through a Graphical User Interface (GUI). The server processes user requests and manages DSS components, while an external containerized service handles the deployment and execution of AI pipelines. The final pipeline configuration is presented to the user based on the processed data and system interactions.

  - **Compatibility**: Compatibility with major web browsers.

- **Architecture**: The architecture of DSS4EX follows a **client-server model**, where a centralized DSS web server manages AI pipeline configurations and interacts with users via a graphical interface. Additionally, the system integrates an **external containerized service** responsible for deploying and executing AI pipelines. The architecture consists of three main components:

  - **Client Side**: Users interact with DSS4EX through a web-based Graphical User Interface (GUI). The client allows users to configure AI pipelines, visualize processing steps, and analyze results interactively.
  - **Server Side**: The DSS web server processes user requests, manages configurations, and interacts with the external container. It oversees data handling, model selection, pipeline modifications, and the orchestration of AI tasks.
  - **External Pipeline Deployment**: An external containerized service is used to deploy and execute the configured AI pipeline. This ensures modularity, scalability, and efficient computation for training and inference tasks.

  o **Software Network-based Architecture**: Figure 3.3 illustrates the network-based architecture. DSS4EX follows a client-server paradigm, where users interact with the DSS via a graphical web interface. The DSS server processes configuration requests and manages AI pipeline structures. The final execution of the pipeline is offloaded to an external containerized service, which independently processes the AI pipeline and returns the results.

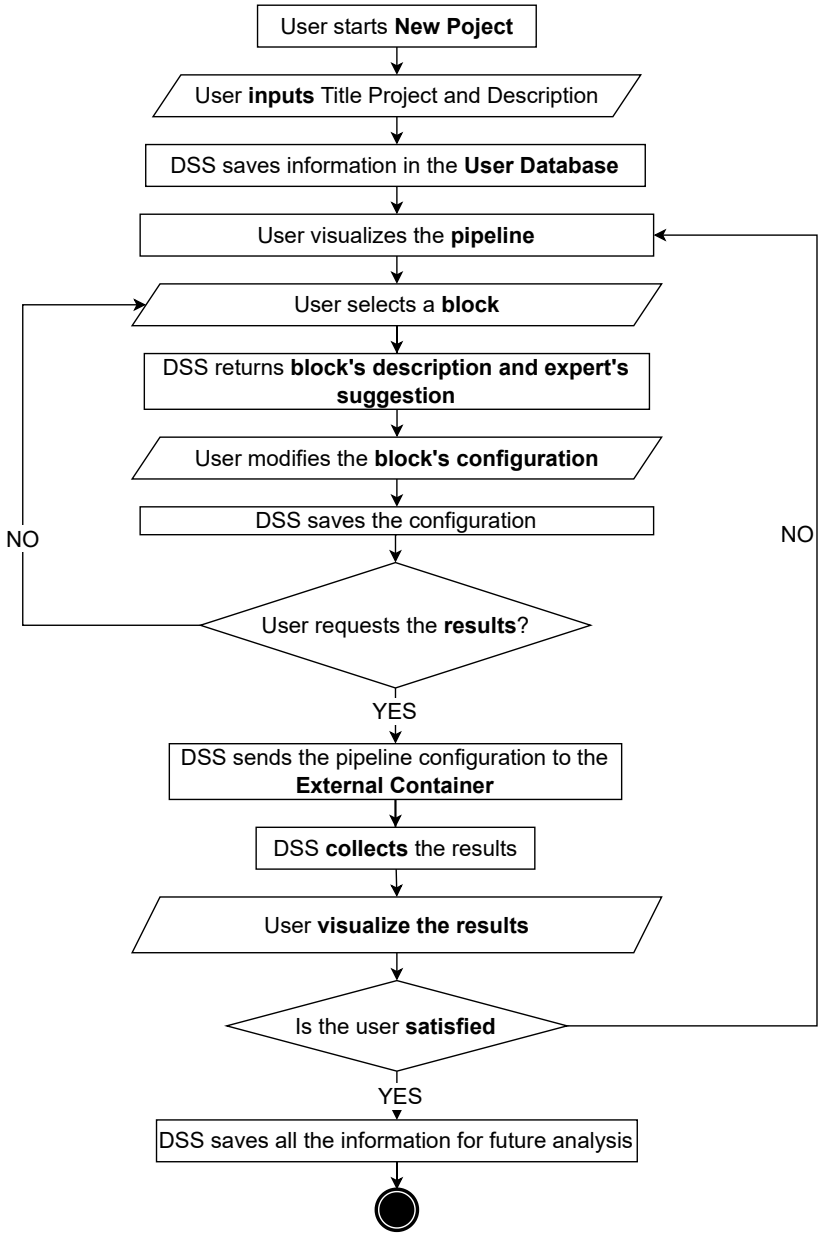


Figure 3.4. **Processing Flow of DSS4EX.** This diagram outlines the structured interaction between users, the DSS web server, and the external containerized service. The external container is responsible only for deploying the AI pipeline and returning the computed results. However, the DSS4EX software demo extends beyond execution, allowing users to compare multiple pipeline configurations and analyze performance differences.

- **Processing Flow**: The DSS4EX framework follows a structured processing flow divided into three main phases. Figure 3.4 illustrates the logical flow of user interaction with DSS4EX.
  1. **Project Initialization Phase**: Users create a new project by entering a title and description. The system saves this metadata in the User Database.
  2. **Pipeline Configuration Phase**: Users interact with the Graphical User Interface (GUI) to visualize and modify the AI pipeline.
     - Users select a pipeline block to inspect.
     - The DSS retrieves the block's description and provides expert suggestions.
     - Users modify the block's configuration if needed.
     - The DSS saves the updated configuration.
  3. **Execution and Analysis Phase**: Users request pipeline execution, triggering the following steps:
     - DSS4EX sends the pipeline configuration to an External Container for deployment and execution.
     - The external service processes the AI pipeline and returns the computed results.
     - Users visualize the execution results.
     - If the results are unsatisfactory, users can modify the pipeline and re-run the process.
     - If satisfied, DSS4EX saves all configurations and results for future reference.

  In addition to pipeline execution, DSS4EX provides functionalities for comparing different AI pipeline configurations. Users can analyze and contrast performance metrics, leveraging visualization tools to determine the best configuration. This capability enables deeper exploration and evaluation of AI models, making DSS4EX more than just an execution framework—it is also a decision support system for model optimization.

- **Main User Stories**: Below, there are 2 user stories where target clients describe the actions that they need the software to perform.
  - **User Story 1**: *As a Data Scientist, I want to create a new project to explore different AI pipelines so that I can customize configurations and analyze results.*
    - **Acceptance Criteria**

        - The system must allow users to create and save new projects.
        - Users should be able to define project titles and descriptions.
        - Initial configurations must be set through a guided interface.
- **User Story 2**: *As a data scientist, I want to visualize and adjust the AI pipeline's structure so that I can understand and improve the model setup.*
    * **Acceptance Criteria**
        - Users must be able to view an interactive representation of the AI pipeline.
        - The system should provide expert suggestions for each configurable component.
        - Users must be able to enable, disable, or modify pipeline components.
- **User Story 3**: *As an AI enthusiast, I want to explore and understand the Decomposition-Residuals Deep Neural Network (DR-DNN) pipeline so that I can gain insights into how AI models work and improve my knowledge of time series forecasting.*
    * **Acceptance Criteria**
        - The system must provide an intuitive visualization of the DR-DNN pipeline structure.
        - Step-by-step explanations should be available to describe each component of the pipeline.
        - Expert-guided insights should be integrated to explain hyperparameters and their effects.
        - Interactive tooltips or pop-up descriptions should be provided for key elements.
        - A demo mode should allow users to test different configurations and see how changes impact the results.

3.5.2 DSS4EX Real-Life implementation as software demo

This subsection describes a software demo that validates the effectiveness of the DSS4EX framework. It is divided into three subsections: the context, the description of the pipeline, and the dataset.

The Context: Software Demo's Background

The framework is designed to help users understand, modify, and/or deploy a complex pipeline for time series forecasting. In other words, providing that

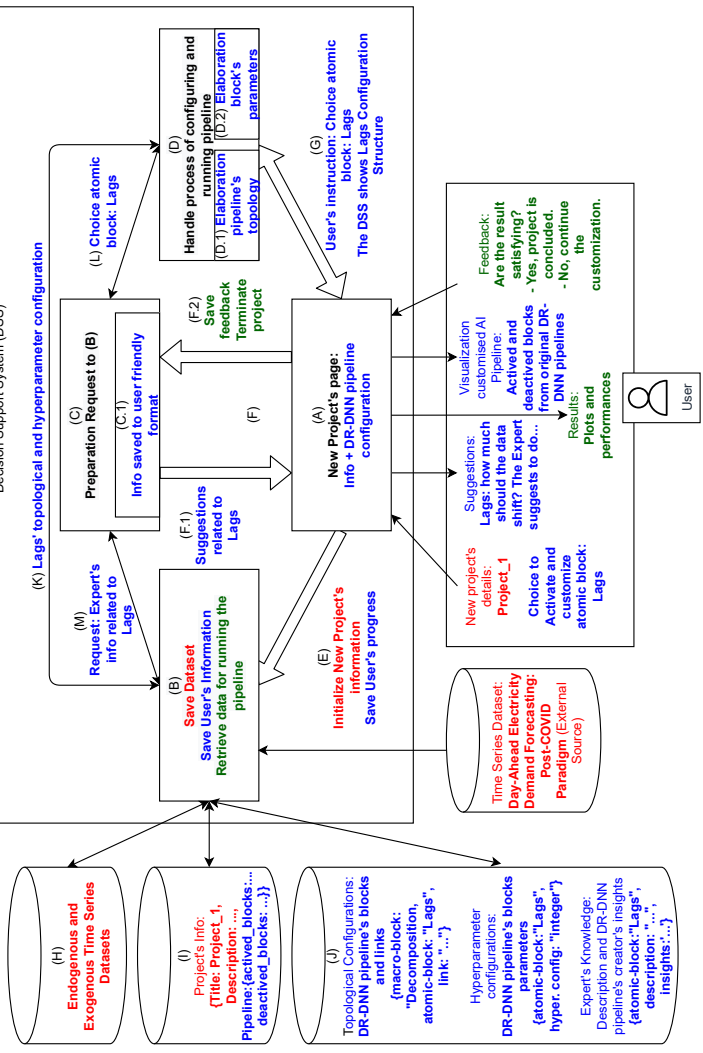


Figure 3.5. **Example of DSS4EX interaction in the DR-DNN use case.** The image shows the flow of information for initializing a new project (red), selecting Lags blocks (blue), and training/visualizing results (green). The user creates a project via the interface (A), which is saved in the database (I) through the Data Management subsystem (B). To visualize the pipeline, the interface requests topological configuration from the Model Management subsystem (D) (G). When modifying the Lags block, the system communicates with the Hyperparameter Entity (D.2) and Knowledge Management subsystem (C) for suggestions. Finally, the user triggers the Model Management subsystem (D) to apply the configuration to the time series dataset.

a researcher who has designed a complex pipeline for time series forecasting aims to show, spread, and promote his/her work, he/she can use the framework DSS4EX to develop a software tool that will show his/her product in a more accessible manner to a wider audience that includes not only a more experienced public but also a less knowledgeable audience. After the tool has been developed, an interested user can interact with the software to gain an understanding of the latest approaches to tackling time series forecasting problems. In this interaction, the user can visualize the pipeline, adjust parameters, and access the designer's insights through a practical application, all without the necessity of coding. Since DSS4EX is presented as a prototype, this demo serves primarily as a proof-of-concept, demonstrating the framework's potential for usability and accessibility rather than as a finalized product for large-scale use.

Consequently, the software tool is used by a user to understand and navigate through the pipeline created by the designer.

As described in Section 3.5.2, the chosen pipeline to integrate into the software demo is the Decomposition Residuals Deep Neural Network (DR-DNN) designed by Theodorakos et al. (2022) [102] and the real-life problem is to predict the electricity demand for the following day.

The Artificial Intelligence Pipeline for Time Series Forecasting

The target problem described in Section 3.5 is time series forecasting. So, the goal is to analyze historical and current time series data and, based on it, predict future target values [62, 55]. Many pipelines are available in the literature [62, 55], and in this chapter, the architecture described in Theodorakos et al. (2022) [102] was selected to be integrated into the DSS design. It was chosen because the pipeline, called Decomposition-Residuals Deep Neural Network (DR-DNN) and shown in Figure 3.6, offers a good level of complexity to display in the DSS. For instance, it has a consistent number of steps that the DSS can help understand and support during the tuning process. The pipeline is designed to predict the following 24 hours of electricity consumption using a dataset comprising the power load to predict and additional external variables.

The DR-DNN pipeline is composed of two layers:

1. **The Decomposition Residuals layer (DR)** processes endogenous and exogenous time series data by using different statistical techniques in order to extract new time series signals logically. The endogenous signal represents the variable to predict, for instance, the power load. The exogenous signals are external variables, such as weather data, that can help the forecasting model.

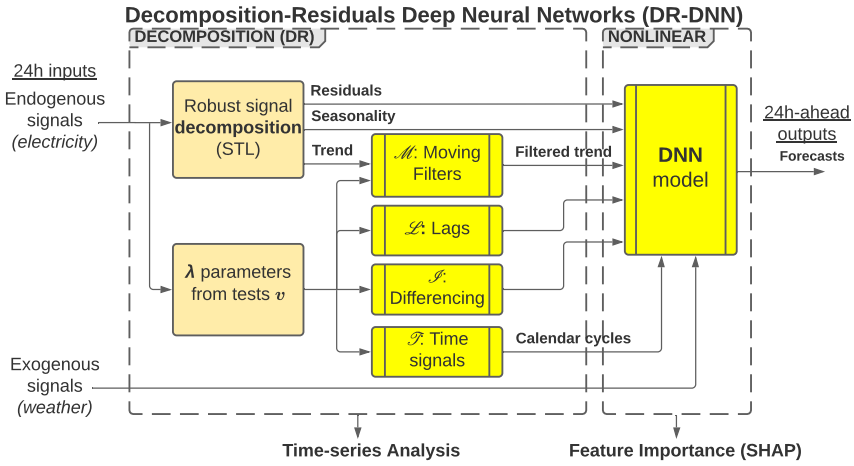


Figure 3.6. **The Decomposition Residuals Deep Neural Network (DR-DNN) pipeline proposed by Theodorakos et al. (2022) [102].** It is a two-layer pipeline. The first part, the Decomposition Residuals layer, extracts information (trend, seasonalities, and residuals) from the endogenous signal (the variable to be forecasted), which, in this case, corresponds to the electricity consumption. The second part consists of the Non-linear layer, which takes the signals extracted from the Decomposition Residuals layer along with the exogenous signals (independent external variables, like weather, precipitation, etc.) as inputs. It then generates the forecast for electricity consumption as its output. Model explainability is possible via: (1) time series analysis (using classical system identification tools) and (2) ranked input feature importance (using the game-theoretic Shapley values).

2. **The Non-linear layer** inputs the exogenous signals and the DR layer's output into a Deep Neural Network (DNN) model. Once the model training phase has been completed, the model will be able to predict the following 24-hour period of the endogenous signal. The use of SHapley Additive exPlanations (SHAP) aids users in interpreting the trained models (knowing that DNNs are essentially black box models) by highlighting the endogenous or exogenous inputs that have the greatest influence on the forecasts.

The Dataset

The dataset used in the software demo is the same one as the Chapter introducing the DR-DNN Pipeline [102]. It is a public time series dataset titled "Day-Ahead

Electricity Demand Forecasting: Post-COVID Paradigm" [6]. This dataset, already pre-processed to ensure high data quality, focuses on evaluating the framework's effectiveness without data quality issues. The dataset spans from March 18, 2017, at 8 a.m., to February 16, 2021, at 11 p.m., covering approximately four years of hourly data and totaling 34,360 samples. Table 3.1 shows a segment of the dataset. It consists of 8 columns representing both endogenous and exogenous variables. The primary endogenous variable, "Load (kW)," corresponds to the city power load that the pipeline aims to predict. The exogenous variables, such as "Humidity," "Wind Direction," and "Wind Speed," among others, provide additional information to improve the machine learning model's forecasting accuracy. This dataset was critical for evaluating the impact of COVID-19-related demand shifts on electricity consumption. The competition aimed to develop and disseminate state-of-the-art load forecasting techniques to mitigate the adverse effects of pandemic-related demand uncertainties, providing a robust testing ground for advanced forecasting methods.

3.5.3 Software Demo Description

The DSS4EX software demo provides a practical demonstration of how the framework can be applied to electricity demand forecasting using the Decomposition Residuals Deep Neural Network (DR-DNN) pipeline. This implementation showcases the core functionalities of DSS4EX, allowing users to configure AI pipelines, modify parameters, and analyze results through an intuitive and interactive interface.

Figure 3.5 maps the framework's workflow to this specific use case, highlighting the key steps of user interaction:

- **Project initialization (red)**: where users create and define a new forecasting project,
- **Pipeline configuration (blue)**: allowing modifications of AI pipeline components, such as the "Lags" block,
- **Pipeline execution and result visualization (green)**: where users run the configured pipeline and analyze the forecasted results.

To provide a clearer understanding of the software functionality, Figures 3.8, 3.9, 3.10, 3.11, and 3.12 showcase different interface components, guiding users through the project creation, pipeline modification, training execution, and

[6] `https://ieee-dataport.org/competitions/day-ahead-electricity-demand-forecasting-post-covid-paradigm`

Time	Load (kW)	Pressure_kpa	Cloud Cover (%)	Humidity (%)	Temperature (C)	Wind Direction (deg)	Wind Speed (kmh)
2017-03-18 08:00:00	1060602	1011,869	44	77,41667	15,83188622	268,375	5,28125
2017-03-18 09:00:00	1141663	1013,41	44	74,33333	19,36938622	289,375	7,13375
2017-03-18 10:00:00	1195235	1013,54	44	74,33333	21,59438622	285,625	14,72375
2017-03-18 11:00:00	1208175	1013,086	44	76,25	22,38688622	279,375	16,6312
2017-03-18 12:00:00	1216409	1013,589	44	77,91667	21,39188622	270,75	13,36375
2017-03-18 13:00:00	1211772	1013,695	44	76	22,12938622	282	16,0625

Table 3.1. **Snapshot of the dataset used in the software demo.** It is a public time series dataset called "Day-Ahead Electricity Demand Forecasting: Post-COVID Paradigm" [38]. The time series dataset starts on the 18th of March 2018 at 8 a.m. and ends on the 16th of February 2021 at 11 p.m., and the signals are collected hourly. It comprises the endogenous signal "Load (kW)," the city power load, and the exogenous signals "Pressure kpa," "Cloud Cover," "Humidity," "Temperature," "Wind Direction," and "Wind Speed."

final result visualization. The "Lags" block serves as an illustrative example throughout this section, demonstrating how DSS4EX allows users to modify AI configurations with expert guidance.

For a more immersive demonstration, Figure 3.7 provides an embedded video walkthrough of the DSS4EX software, offering a step-by-step overview of its functionalities. The video is also accessible on YouTube at `https://youtu.be/1d4e9WgE0g4`, enabling users to observe the full process in action.

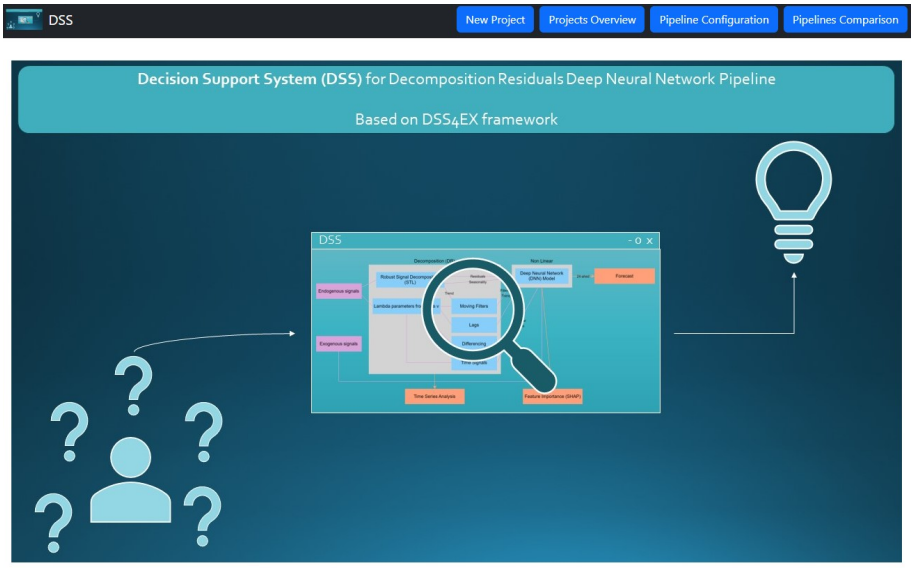


Figure 3.7. **Software Demo Video** This video demonstrates the core functionalities of the software prototype developed using the DSS4EX framework. (Access the video here: `https://youtu.be/1d4e9WgE0g4`)

New Project

The first user-system interaction is the initialization of a new project. The interface (Fig. 3.5A) shows the New Project Page (Fig. 3.8) where the user can indicate a unique title and a short description of the configuration of the pipeline. As an example, two new projects were created:

1. Title: "Pipeline_without_DR" and Description: "The pipeline is configured deactivating the decomposition residuals blocks."
2. Title: "Pipeline_DR-DNN" and Description: "The pipeline is configured using the designer's suggestion."

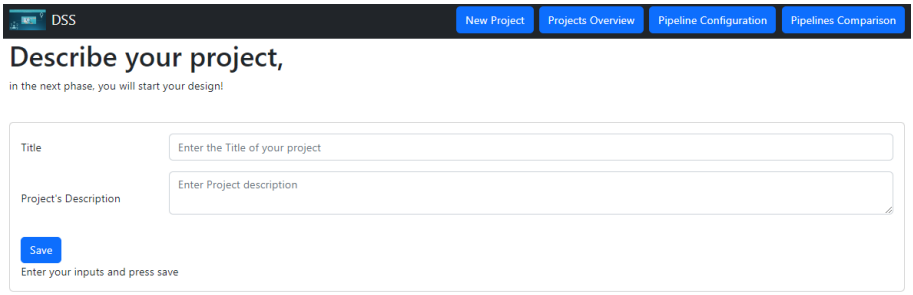


Figure 3.8. **New Project Page Interface.** This screenshot displays the initial interface where the user can enter a unique title and description for a new project. The user interacts with this page to create and configure a new pipeline project within the DSS4EX framework.

After saving this information (Fig. 3.5A), the interface triggers the Intake Flow (Fig. 3.5E) that sends it to the Data Management subsystem (Fig. 3.5B). This stores the title and the description in the User's Database (Fig. 3.5I) as a new project.

Pipeline Visualization

After the new project information is saved, the interface (Fig. 3.5A) shows the list of projects (Fig. 3.9), where the user can visualize all the projects initialized or completed. In this software demo, the pipeline associated with each project was designed by Theodorakos et al. (2022) [102], and it is the only one available. As described in Section 3.5.2 and shown in Figure 3.6, the pipeline has two layers. The first layer, Decomposition Residuals (DR), tries to extract as much information as possible from the data, generating new signals that are fed to the second layer, the Deep Neural Network (DNN). The training of the model goes as follows: the DNN model reads the previous 24 hours of data (one day), then tries to predict the next 24 hours, compares the output with the actual data, and learns from the process. This operation is executed sequentially on the entire training dataset. After the training phase has finished, the model's accuracy is assessed on a test dataset that was completely unseen by the model. The type of DNN model used in the software demo is a Long Short-Term Memory (LSTM) recurrent neural network.

Pipeline Configuration

After initializing the project and visualizing the pipeline, the user can access the Configuration Page (Fig. 3.10), where it is possible to read the description

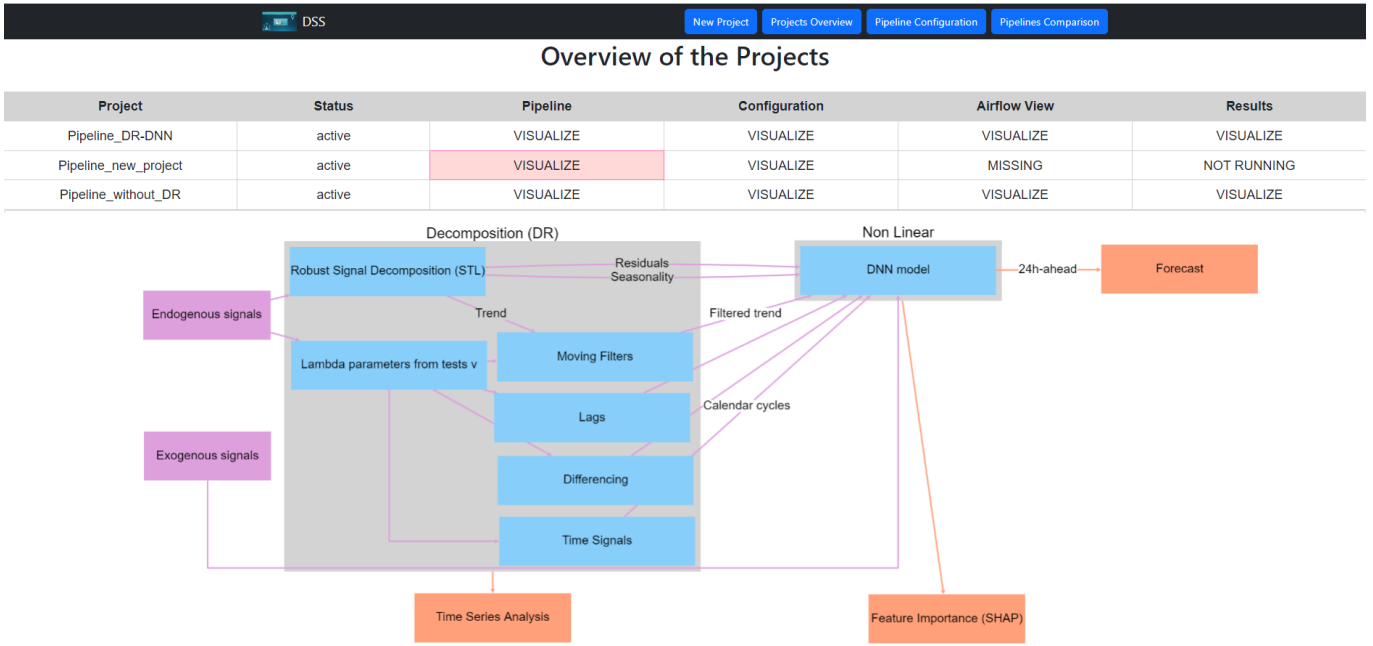


Figure 3.9. **Project List and Pipeline Visualization Interface** This screenshot displays the project list page where users can view and select their initialized or completed projects. For each project, the associated pipeline designed by Theodorakos et al. (2022) [102] is visualized, with detailed insights into its two-layer structure: Decomposition Residuals (DR) and Deep Neural Network (DNN). The interface allows users to explore and manage their projects within the DSS4EX framework.

of each block, receive the designer's suggestion, and decide about configuration parameters. The description and the required configuration are retrieved by the Model Management subsystem (Fig. 3.5D) through the Data Management subsystem (Fig. 3.5B), which takes this information from the database (Fig. 3.5J). In detail, the Topological Entity (Fig. 3.5D.1) will retrieve and collect the structure of the pipeline, instead of the Hyperparameter Entity (Fig. 3.5D.2), which will handle the parameter requirements and verify whether the selected value for a specific parameter adheres to the expert's rules. The Table A.1 in the Appendix A lists all the blocks of the pipeline as well as their descriptions. The Knowledge Management subsystem fetches the suggestion related to a block (Fig. 3.5C) stored in the database (Fig. 3.5J), as well through the Data Management subsystem (Fig. 3.5B). The suggestions are based on the expert's insights (Fig. 3.5J), which are static texts, and they are sent to the User Interface (Fig. 3.5A) by the Suggestion Segment ((Fig. 3.5F.1) for this software demo. Figure 3.10 offers an example of the configuration: the user intends to configure the block "Lags." First, the user can enable or disable this block based on the necessity of having the lag operation. In the DR-DNN pipeline, only the "Endogenous signal" and "DNN Model" blocks are mandatory. The former is crucial as it signifies the model's prediction target, and the latter is necessary for the pipeline's execution. The block "Lags" only has a single parameter, the number of lags, which is an integer indicating how many time steps the endogenous signal should be shifted. For this particular example, the designer's suggestion is to set the number of lags to 1. At this point, the user is free to decide whether to set this parameter manually (e.g., following the designer's suggestion) or to use the value provided by "λ parameters from tests v" block (e.g., by computing the partial autocorrelation). The configuration is saved by the Model Management subsystem (Fig. 3.5D) through the Data Management subsystem (Fig. 3.5B) into the database (Fig. 3.5J). Table 3.2 and Figure 3.13a show, respectively, the values of the main parameters and the pipeline with the deactivated blocks, colored in gray, for the previously defined project "Pipeline_without_DR." Table 3.3 and Figure 3.15a depict the same information but for the project "Pipeline_DR-DNN."

Pipeline Training

After configuring all the blocks, the user triggers the Analysis Flow (Fig. 3.5G) that activates the Model Management subsystem (Fig. 3.5D) to run the pipeline through the interface (Fig. 3.5A). As in the case of this software demo, the Model Management subsystem (Fig. 3.5D) can be connected to a separate service that handles the actual computation in order to improve efficiency and reduce execution time. This means that the Model Management subsystem (Fig. 3.5D) needs to retrieve the configuration and prepare it for the external

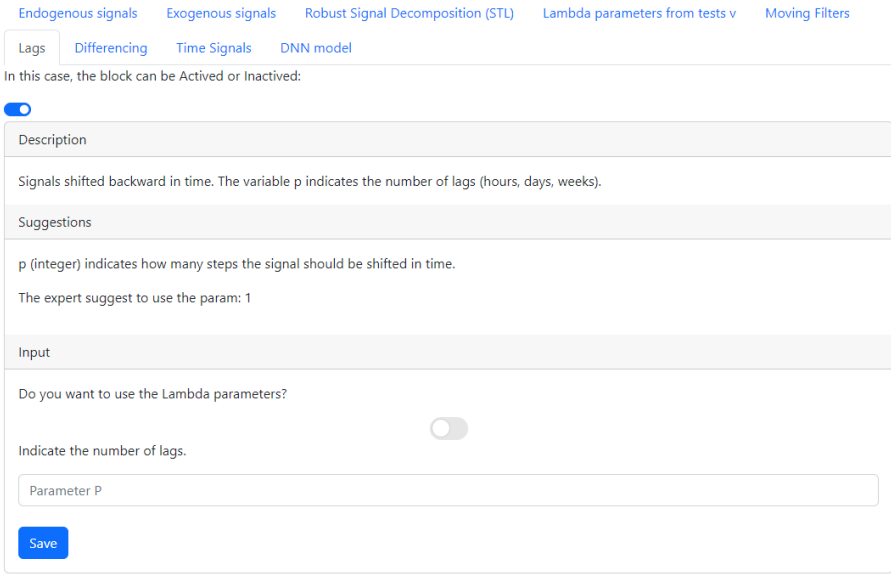


Figure 3.10. **Pipeline Configuration Interface.** This screenshot illustrates the Configuration Page, where users can read detailed descriptions of each block, receive the designer's suggestions, and configure the block parameters. In the image, it is highlighted the configuration process for the "Lags" block in the DR-DNN pipeline, where users can enable or disable the block and adjust the number of lags based on the designer's recommendations.

Projects	**Train/Val./Test**	**Normalization**	**Exogenous**
Pipeline_without_DR	70/20/10	YES	All
Pipeline_DR-DNN	70/20/10	YES	All

STL/parameter(period)	**λ Test**	**Moving Filter**	**Lags**	**Differencing**
Skipped	Skipped	Skipped	Skipped	Skipped
Yes/7	Skipped	2	1	0

Table 3.2. **Comparison of Configuration Parameters.** This table compares the parameter values of two projects, Pipeline_without_DR and Pipeline_DR_DNN, highlighting the differences in configuration, such as the activation of specific blocks and their parameter settings.

service. It also recovers the dataset from the database (Fig. 3.5H) through the Data Management subsystem (Fig. 3.5B). In the software demo, if the user wants to supervise how the training of the pipeline on the external service is going, they can access the dedicated page through the interface (Fig. 3.5A), as shown in Figure 3.11.

In the software demo, the external service begins processing the full pipeline

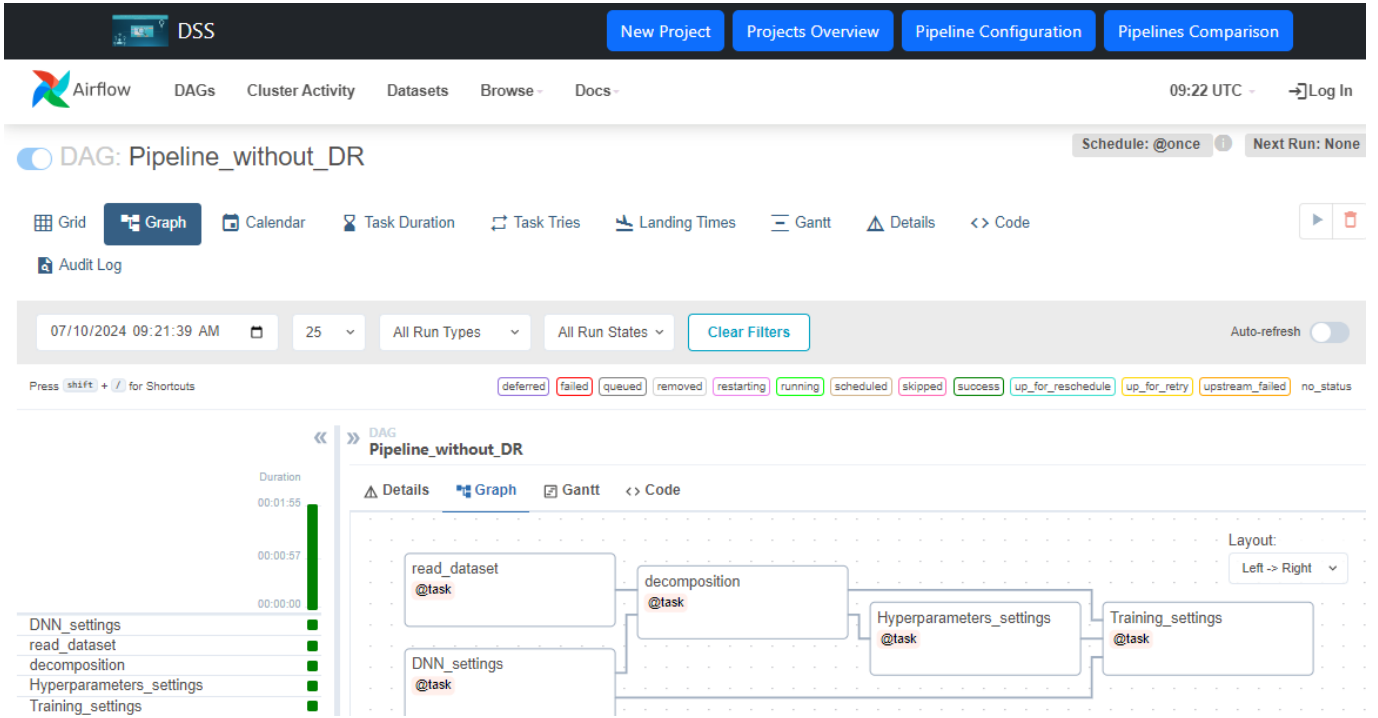


Figure 3.11. **Pipeline Training and External Service Monitoring.** This screenshot shows the interface where the user can monitor the training process of the AI pipeline. The Model Management subsystem triggers the execution on an external service, and the interface provides real-time updates on the training status. The visualization helps users oversee the progress and time efficiency of the pipeline.

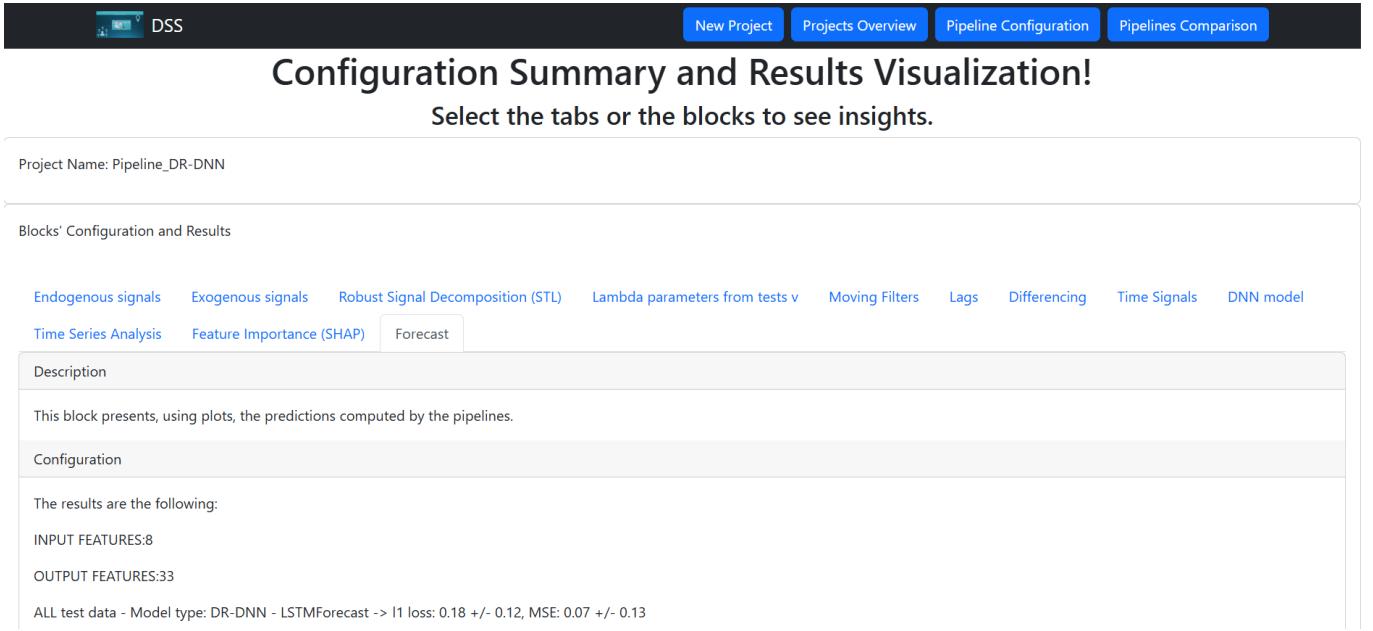


Figure 3.12. **Results Visualization Page.** This figure showcases the results page where the user can visualize the pipeline's outcomes. It includes tabs for time series analysis, feature importance, and forecasting, allowing the user to explore different aspects of the pipeline's performance.

Hidden Size	Output Size	Batch Size	Learning Rate	Epochs
64	24	32	0.001	20

Table 3.3. **Hyperparameter Table for Pipeline Configuration** This table presents the hyperparameter values for the selected pipeline configuration, detailing the adjustable parameters for each block.

configuration in less than 4 minutes. This **execution time** refers to the total time required by the system to process all configured blocks in the pipeline, including data retrieval, preprocessing, configuration loading, model training, and result visualization, and to return the outputs to the user interface. When only the essential blocks are active (i.e., the Endogenous System block, the Exogenous System block, and the DNN Model), the execution time is approximately 3 minutes. This timing remains relatively stable even when more blocks are enabled, as the Decomposition Residuals phase introduces minimal computational overhead.

It is important to emphasize that the **training phase** of the Deep Neural Network (DNN) model is part of the overall execution flow and represents the most computationally intensive step. However, training is not measured separately, as the system reports only the complete execution time for running the configured pipeline. Additionally, due to the stochastic nature of model initialization, slight variations in runtime are expected. In the examples tested, the “Pipeline_without_DR” completed execution in 1 minute and 55 seconds, while “Pipeline_DR-DNN” required 2 minutes and 17 seconds.

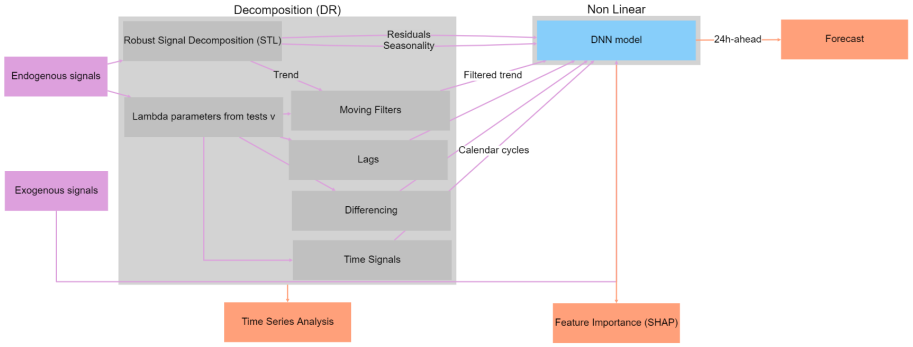
Pipeline Evaluation and Results

After the pipeline is trained, the Model Management subsystem (Fig. 3.5D) collects the results and saves them through the Data Management subsystem (Fig. 3.5B) into the database (Fig. 3.5I). When the user requests the results, the Model Management subsystem (Fig. 3.5D) retrieves and sends them to the User Interface (Fig. 3.5A). An example of how the Result page looks like is shown in Figure 3.12. Here, there is a tab for each block in the pipeline. Under the "Time Series Analysis," "Feature Importance (SHAP)," and "Forecast" tabs, users can visualize the pipeline’s results, while for the other tabs, only a summary of the configuration is available. "Time Series Analysis" incorporates the description, snapshots, and information plots related to the dataset analysis, which can be downloaded. The "Feature Importance (SHAP)" shows plots using Shapley Additive Explanations to explain the output of the machine learning models. The plot illustrates the features with the

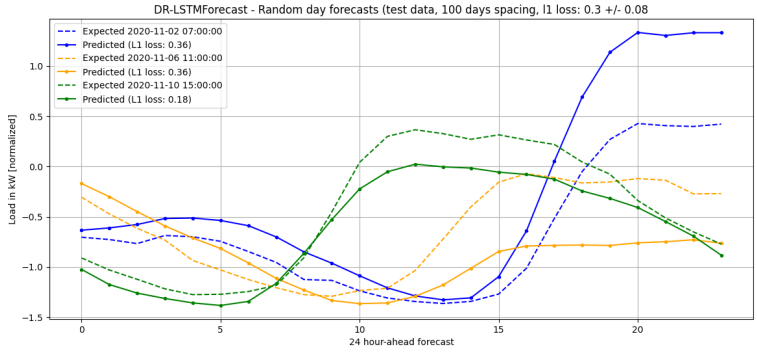
most substantial impact on the model training. Examples of such plots are depicted in Figures 3.14 and 3.16 that represent the SHAP plot obtained by the training of "Pipeline_without_DR" and "Pipeline_DR-DNN," respectively. In the tool, an expert's comment is displayed, explaining that the color represents the contribution of each feature at specific forecast hours. In the models under consideration, "Load(kW)_trend_23h" emerges as the most influential feature. However, an interesting observation is that in the first case (Fig. 3.14), only the load and a single exogenous signal, "Temperature (C)," appear as significant features. In contrast, with the integration of the decomposition residual blocks, Fig. 3.16 reveals a more diverse set of feature contributions, utilizing different types of signal decompositions and additional exogenous signals. The last Section containing results is the "Forecast" block, which shows information related to model performances. The model and the predictions can be downloaded for further investigation by the user. The user can visualize the prediction plots of three days, selected randomly but equally for all pipelines, where the solid line represents the prediction, and the dotted line represents the expected values. This allows for an immediate assessment of model performance. Figures 3.13b and 3.15b show the plots generated by "Pipeline_without_DR" and "Pipeline_DR-DNN," respectively. Important to note that the experiments described in this manuscript with the software demo aimed not to duplicate the results of Theodorakos et al. (2022) [102], but to demonstrate the pipeline's basic functionality without delving into the specifics of implementation. When the operation is concluded, and the user is satisfied, the project can be deactivated from the Overview Page (Fig. 3.9), triggering the Feedback Segment (Fig. 3.5F.2) and indicating to the system and the Knowledge Management subsystem (Fig. 3.5C) that the project is finished.

Pipelines Comparison

The Knowledge Management subsystem (Fig. 3.5C) and the Model Management subsystem (Fig. 3.5D) can be extended to support new functions in the software application. For example, in the software demo, comparing two pipelines with the same train/validation/test configuration and the same output size is possible. The Knowledge Management subsystem (Fig. 3.5C) and the Model Management subsystem (Fig. 3.5D) work to retrieve and compute the necessary information using results and insights fetched by the Data Management subsystem (Fig. 3.5B) from the database (Fig. 3.5I) and showed through the User Interface (Fig. 3.5A). Figure 3.17 and Figure 3.18 show how the Comparison page looks like. In general, the page presents the predictions generated from the two pipelines, comparing them with the expected values in a plot where the user can select different time slots. Additionally, based on the same time slot, it is possible to visualize the plot of the residuals, calculated as the difference between the



(a) Pipeline_without_DR overview after configuration.



(b) Forecasting Results for Pipeline_without_DR.

Figure 3.13. **Overview of the pipeline, results, and SHAP plots of the project Pipeline_without_DR.** Image (a) shows the pipeline after configuration, with grey blocks representing deactivated components related to the DR layer. Image (b) displays the model's predictions alongside expected values for three randomly selected days.

expected and actual values. To complete the comparison, some metrics are calculated for each project on the selected time slots. In particular, Figure 3.17 and Figure 3.18 present a detailed comparison of the two pipelines described before, for the time period '2020-09-28 18:00:00.000" - "2020-09-29 02:00:00.000."

Figure 3.17 presents the top of the Comparison page. It includes the selection of the time slot and the first pipeline that compares the expected values with the prediction calculated by the two pipelines. Figure 3.18 depicts the bottom of the Comparison page. It comprises the residual plot for the two pipelines and a table with error metrics, such as Mean Squared Error (MSE), Mean Absolute Error (MAE), etc. In the software demo, the Knowledge Management subsystem (Fig. 3.5C) provides static descriptions to help the user understand

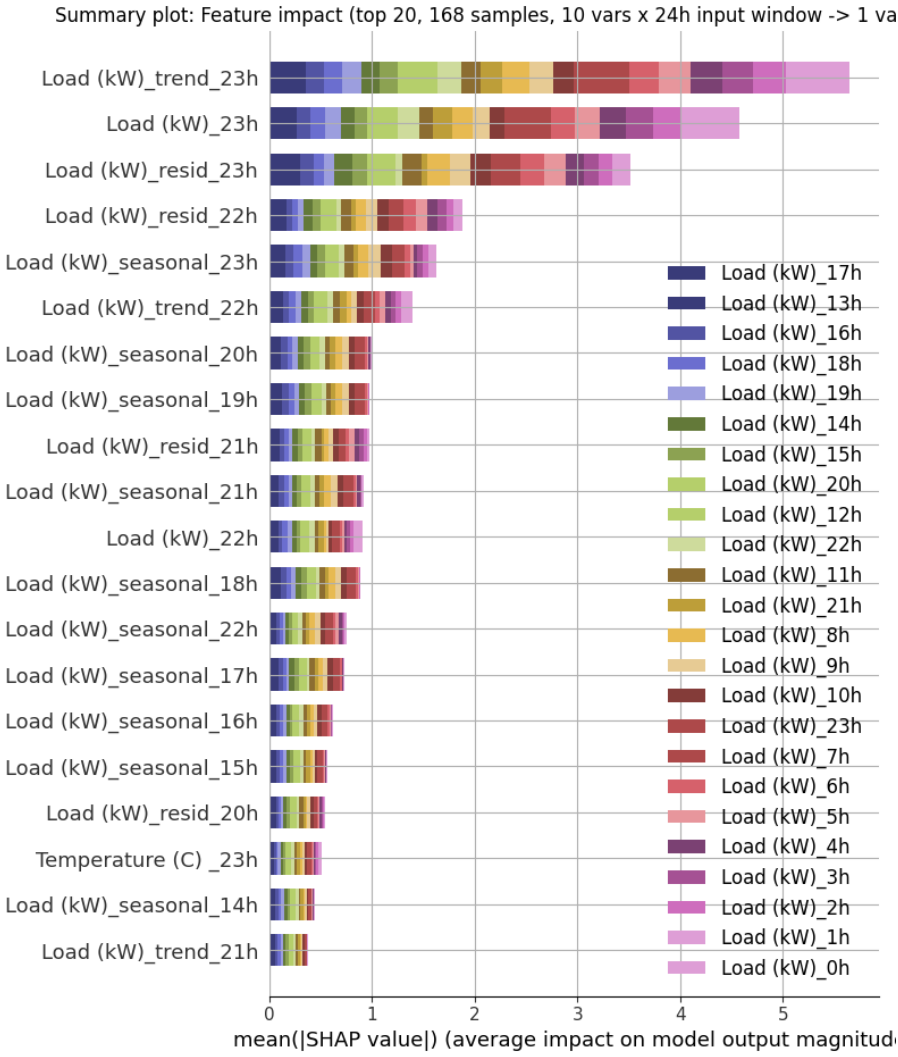
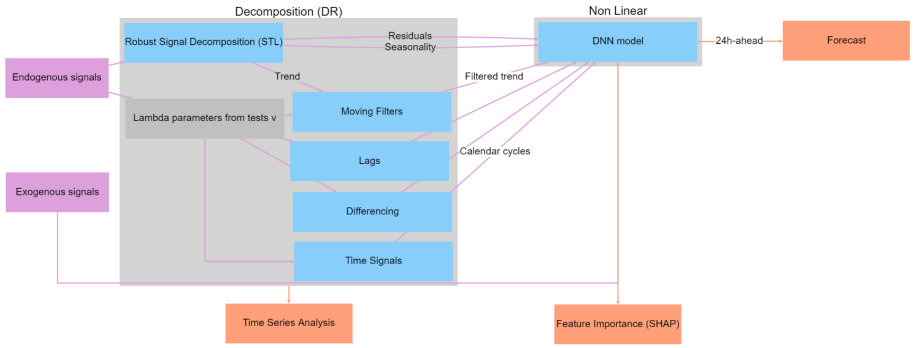
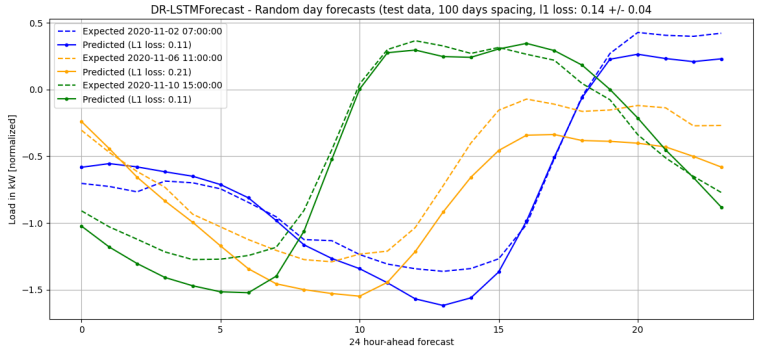


Figure 3.14. **SHAP Plot for Pipeline_without_DR.** This plot highlights the ranking of the most significant features using feature importance. In this scenario, the SHAP plot indicates that features related to "Load(kW)" dominate as significant, with "Temperature (C)" being the sole exogenous variable of importance. An embedded expert comment interprets this limited feature set, noting a primary focus on load and one exogenous signal. The comment further suggests that incorporating additional external signals could improve the model's robustness.



(a) Pipeline_Dr-DNN overview after configuration.



(b) Forecasting Results for Pipeline_DR-DNN.

Figure 3.15. **Overview of the pipeline, results and SHAP plots of the project Pipeline_DR-DNN.** Image (a) presents the pipeline after configuration. The grey block, specifically the "λ parameters from tests", is the deactivated block. Image (b) plots the predictions found by the model with their expected values for three randomly selected days.

the plots and the metrics.

3.5.4 Implementation Details

- **Technology Stack** The following technologies were used for the implementation of the software demo. Figure 3.19 illustrates the relationship between the system architecture, as described in Section 3.5.1, and the technologies summarized in this section.

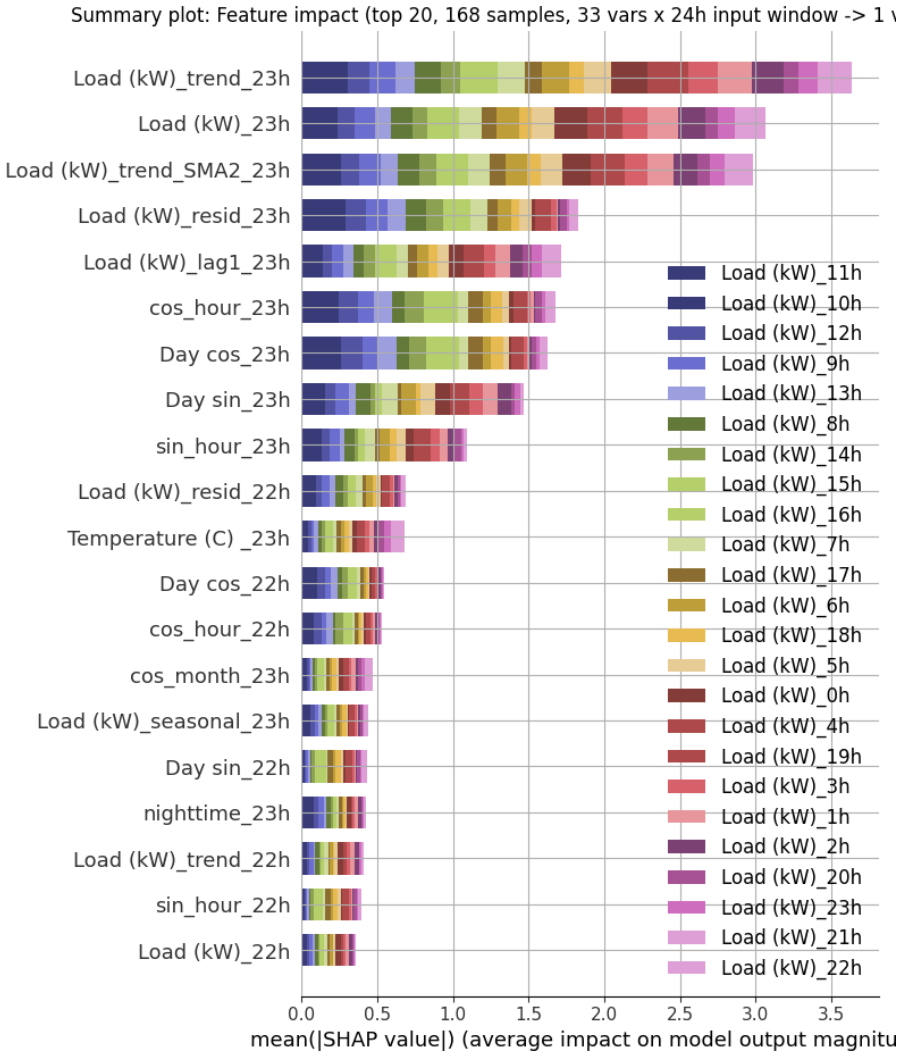
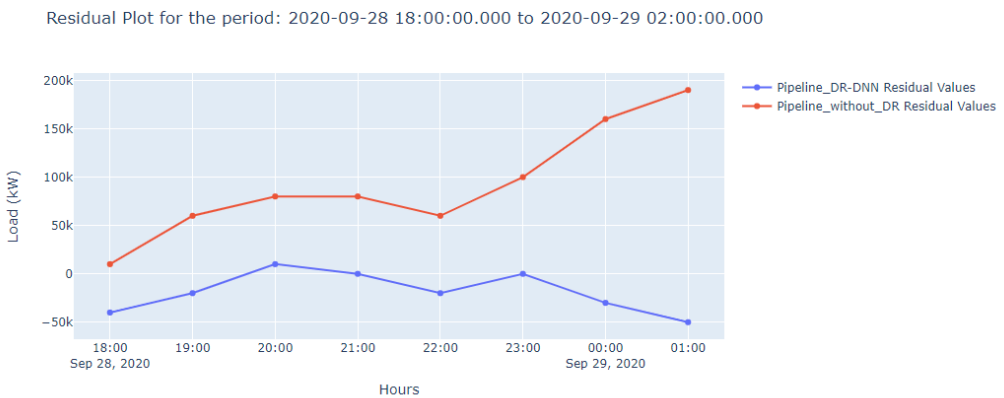


Figure 3.16. **SHAP Plot for Pipeline_DR-DNN.** This plot This SHAP plot reveals the most influential feature and a more complex feature interaction, where a range of signals, including multiple types of signal decompositions and additional exogenous variables, are used. The tool provides an expert comment here as well, which guides users by explaining that the decomposition residual block allows the model to capture more nuanced influences from diverse features, thus enhancing prediction accuracy. "Load(kW)_trend_23h" emerges as the most influential feature.



Figure 3.17. **Details of the Comparison Page: Top View** The image displays the top of the web page. It includes the selection of the time slots and plot of the predictions obtained by running Pipeline_without_DR and Pipeline_DR-DNN with the expected values.



The table shows 4 metrics on the residual (expected value - actual value):

1) Mean Absolute Error (MAE) measures errors between paired observations expressing the same phenomenon.

2) Mean Squared Error (MSE) measures the average of the squares of the residual values.

3) Root Mean Squared Error (RMSE) shows how far predictions fall from measured true values using Euclidean distance.

4) BIAS measures the avarage of the residual values.

Metrics	Pipeline_DR-DNN	Pipeline_without_DR
MAE	21250	92500
MSE	737500000	11475000000
RMSE	27156.95122800054	107121.42642814275
bias	-18750	92500

Figure 3.18. **Details of the Comparison Page: Bottom View** The image shows the bottom of the page. It comprises the residual plot, so the difference between the expected values and the predictions calculated by the pipelines, Pipeline_without_DR and Pipeline_DR-DNN. Some error metrics (Mean Squared Error, Mean Absolute Error, etc.) are also calculated to provide more insights. The residuals plot shows that the Pipeline_without_DR over-estimates the forecasted values.

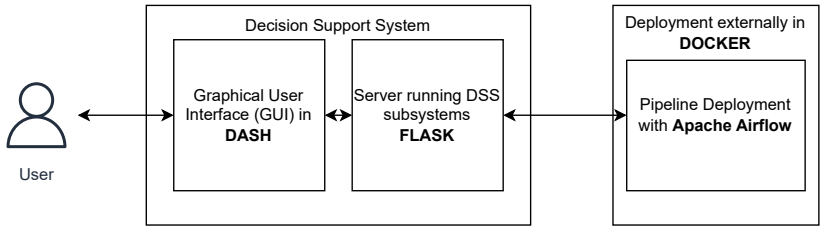


Figure 3.19. **Architecture of DSS4EX** This figure illustrates the technological interactions within DSS4EX. The user interacts with the Decision Support System (DSS) through a Graphical User Interface (GUI) developed in Dash. The GUI retrieves information processed by the server, which hosts DSS components using the web-server Flask. The configured AI pipeline is deployed in a Docker container managed by Apache Airflow. In the end, the user is shown a personalized pipeline resulting from the requests made and based on the retrieved information from the Apache Airflow service.

  - o **Frontend**: Dash[7] (Python-based web framework for UI development).
  - o **Backend**: Flask[8] (Python micro-framework for web services).
  - o **Database**: Relational database (Json File System) for storing metadata, configurations, and results.
  - o **Workflow Management**: Apache Airflow[9] for orchestrating AI pipeline execution.
  - o **Machine Learning Framework**: PyTorch[10] for deep learning model development.
  - o **Containerization**: Docker[11] for deployment and isolated environments.
  - o **Version Control and Code Repository**: The pipeline's source code is available at `https://github.com/temp3rr0r/DR-DNN`.
  - o **Visualization Libraries**: Matplotlib and Plotly for rendering pipeline structures and results.
  - o **Libraries**: Appendix B provides a detailed list of the packages used during development.

- **Test Cases Performed** The system has been tested using the following test cases.

[7] `https://dash.plotly.com/`
[8] `https://flask.palletsprojects.com/en/stable/`
[9] `https://airflow.apache.org/`
[10] `https://pytorch.org/`
[11] `https://www.docker.com/`

- **Project Initialization**: Ensure that new projects can be created, saved, and retrieved correctly.
- **Data Management**: Verify correct storage, retrieval, and preprocessing of time series data.
- **Pipeline Configuration**: Confirm that users can modify, enable/disable, and adjust AI pipeline components.
- **Pipeline Visualization**: Test interactive visual representation of pipeline components and workflows.
- **Training Execution**: Ensure smooth execution of AI model training and validation.
- **Result Analysis**: Validate the accuracy of output visualizations and SHAP plots for model interpretability.
- **Comparison Feature**: Test the ability to compare different AI pipeline configurations and results.
- **Usability and Accessibility**: Conduct user tests to ensure a smooth and intuitive UI experience.
- **Error Handling**: Simulate various user errors and unexpected inputs to test system robustness.

• **Limitations and Constraints**

  o **Incomplete Features**

    * Automated hyperparameter tuning is not yet implemented.
    * Support for additional domains beyond time series forecasting is planned for future versions.
    * Limited explainability features beyond SHAP visualizations.
    * Currently, only single-user interaction is supported; multi-user collaboration features are under development.

  o **Known Issues**

    * The system requires manual configuration of hyperparameters, which may be challenging for non-expert users.
    * Some pipeline components may not be fully optimized for large-scale datasets.
    * Initial loading time of models can be slow depending on the dataset size.
    * Certain complex configurations may lead to execution errors without explicit error messages guiding users to solutions.
    * Limited support for real-time data streaming; batch processing is currently the primary mode.
    * Deployment on cloud infrastructure requires additional configuration steps for compatibility.

3.6 Results and Discussions

This study presents DSS4EX, a Decision Support System (DSS) framework designed to simplify the exploration, configuration, and comprehension of complex AI pipelines by both expert and non-expert users. To demonstrate DSS4EX's functionality, a prototype was developed as a proof-of-concept. This prototype showcases how DSS4EX can provide a structured, intuitive interface for real-world AI applications, in the context of time series energy forecasting.

The development of DSS4EX followed several guiding objectives: establishing clear information flows between system components, creating a user-friendly prototype accessible to both novice and expert users, and integrating expert insights to assist users in configuring and understanding AI pipelines. Drawing on foundational DSS principles, as described in Section 3.2.3, DSS4EX extends traditional DSS structures by introducing a novel information flow and guidance features tailored to AI pipeline exploration, making it distinct from frameworks that focus primarily on structured decision support.

The prototype, detailed in Section 3.5, was applied to explore the Decomposition Residuals Deep Neural Network (DR-DNN) pipeline, an advanced time-series forecasting model proposed by Theodorakos et al. (2022) [102]. The interface, shown in Figure 3.7, showed to be effective in enabling users to interact directly with the pipeline, even if only informal and preliminary tests were performed, offering responsive insights into its components and configurations (Section 3.5). This user-centered design demonstrated its effectiveness in supporting DSS4EX's goal of accessibility, allowing both novice and experienced users to navigate complex AI processes effectively (Section 3.5.3).

Key features of the software demo included model and dataset download options, a modular interface for pipeline adjustments, and pre-configured expert insights on each pipeline block (Fig. 3.10). These features helped users to understand the purpose and function of each component, while the configuration suggestions assisted them in making informed adjustments. Through an informal review, AI experts evaluated the framework, highlighting DSS4EX's accessibility, intuitive layout, and unobtrusive presentation of experts' guidance within the interface. While informal feedback suggests the framework may be suitable for both expert and non-expert users, its usability and adaptability require formal validation through structured testing. Reviewers found the dataset and model download features particularly beneficial for detailed analysis, further validating DSS4EX's capability to support both learning and exploration.

While DSS4EX currently serves as a functional demonstration, future testing with a diverse user base, including both experts and non-experts, will provide data on usability and engagement. The framework will also be tested with

alternative AI pipelines to assess its adaptability and support for varied configurations.

In comparison with established tools like RapidMiner, KNIME and IBM SPSS, DSS4EX is uniquely focused on accessibility for non-expert users and incorporates expert-driven guidance to simplify pipeline exploration. Unlike these traditional tools, which target domain experts, DSS4EX provides user-friendly, guided exploration, making it accessible to a broader audience. Although RapidMiner and KNIME allow for custom pipeline development, they do not offer interactive guidance tailored for non-expert users. Future DSS4EX developments will focus on extending the system's applicability across different domains and supporting flexible, customizable pipeline structures.

DSS4EX's modular design supports scalability, allowing it to handle larger datasets and more complex pipelines as it matures. Technologies aimed at optimizing scalability are being explored to accommodate high volumes of data and multiple users in diverse application contexts.

In summary, results suggest that DSS4EX prototype has the potential to adapt to practical AI applications, using real world data. It could also be argued that DSS4EX can enable non-expert users to engage effectively with complex AI models through its user-guided suggestions and interactive features, while experts benefit from its flexible configurations. Nevertheless, before being stable enough to be applied in contexts where the processing of larger amounts data, further testing and refinement is needed to enhance the system's scalability and robustness, as discussed in Section 3.7.

3.7 Conclusions and Future Works

This chapter describes a framework called Decision Support System for EXploration (DSS4EX), which aims to characterize a Decision Support System (DSS) for exploring Artificial Intelligence (AI) pipelines. DSS4EX combines elements of a DSS with user-focused features, making it easier to navigate, comprehend and adapt AI models, thanks to the human-in-the-loop approach. To examine and demonstrate the framework's usability, a real-life application was selected and implemented as a software demo. The purpose of this demo was to support a user, whether AI knowledgeable or a novice, in exploring a complex AI pipeline called the Decomposition Residuals Deep Neural Network (DR-DNN) Pipeline [102].

DSS4EX builds on traditional DSS structures, enhancing the interoperability of different component systems. This structure makes DSS4EX useful for various

users: non-experts seeking to understand AI and experts wanting to share or explore complex pipelines in a more intuitive environment.

The software demo demonstrated that, in the context of creating and exploring a time series forecasting pipeline, DSS4EX can help reduce users' time spent exploring, executing, and comparing pipelines. Additionally, the framework's flexibility allows it to be adapted to a broader audience, beyond just knowledgeable AI users.

Nevertheless, DSS4EX has certain limitations. It has been applied to a single use case focused on energy time series forecasting. Moreover, the prototype has not undergone extensive real-world testing, necessitating further validation to evaluate user engagement and performance. Non-expert users may face challenges during configuration, highlighting the need for adaptive guidance features, such as a chatbot or AutoML, to enhance support during the setup process.

Future work on DSS4EX will involve expanding its applicability across diverse domains to test its adaptability and effectiveness beyond energy time series forecasting. A key area of exploration includes fields such as stock market forecasting and weather forecasting.

In addition to domain flexibility, the framework could be expanded to include a broader range of pre-defined pipeline structures, allowing users to test and compare various neural network models such as GRU (Gated Recurrent Unit), Transformer-based architectures, and Temporal Convolutional Networks (TCN), alongside LSTM (Long Short-Term Memory).

To improve its analytical capabilities, future versions of DSS4EX could include more advanced features, such as tools for optimizing model parameters (e.g., Bayesian Hyperparameter Optimization) and more elaborate data preprocessing.

Extensive real-world testing is essential to understand DSS4EX's scalability, user engagement, and effectiveness in diverse settings. This testing would reveal improvements needed to handle large datasets and support multiple users, ultimately preparing DSS4EX for both professional and educational applications where users of all backgrounds can benefit from AI pipeline exploration.

Chapter 4

EX-DSS: An Explorative Decision Support System

This chapter is a published paper which has been modified and reprinted with permission. The original version can be found on:

Rinaldi, Giulia; Botman, Lola; Agudelo, Oscar Mauricio; De Moor, Bart "EX-DSS: An Explorative Decision Support System for designing and deploying Smart Plug Forecasting Pipelines"
KMIS 2024 - The 16th International Conference on Knowledge Management and Information Systems, Porto (Portugal), November 2024
DOI: 10.5220/0012925800003838.
https://www.scitepress.org/Papers/2024/129258/129258.pdf

Abstract

Artificial Intelligence pipelines are increasingly used to address specific challenges, such as forecasting smart plug loads. Smart plugs, which remotely control various appliances, can significantly reduce energy consumption in commercial buildings by about 20% when effectively scheduled using AI techniques. Designing these AI pipelines involves numerous steps and variables, requiring collaboration and shared knowledge among designers. A Decision Support System (DSS) can facilitate this process. This chapter introduces the Explorative Decision Support System (EX-DSS), which extends the classical DSS framework. The EX-DSS integrates an Explorative Management Subsystem to provide project-specific recommendations and a Data Quality Module (DQM) to validate user inputs, ensuring clarity and enhancing information sharing. The EX-DSS architecture framework was tested through a software prototype designed to create AI pipelines for forecasting smart plug loads. The study found that using the EX-DSS improves the quality of suggestions, making them more problem-specific and resulting in a more personalized and potentially meaningful user experience, as observed informally during prototype testing, with a significant potential to reduce energy consumption in commercial buildings.

4.1 Introduction

Designing an Artificial Intelligence (AI) pipeline involves multiple steps, from Data Cleaning (DC) to implementing machine learning methods. This process can be complex and time-consuming, especially when trying to find the most efficient combination. In the context of smart plug forecasting, this challenge is significant, as plug loads account for over 40% of total energy consumption in commercial buildings, excluding lighting, HVAC, and water heating [20]. Smart plugs, which remotely monitor and control electrical appliances, can save up to 20% of electricity through effective scheduling, making AI forecasting and scheduling methods promising solutions [8].

A Decision Support System (DSS) can expedite the design of these AI pipelines. A DSS is a flexible software tool that assists in decision-making processes and allows for shareability and reproducibility among users, necessitating collaboration functionalities and ways to evaluate user input.

This chapter introduces an experimental Explorative Decision Support System (EX-DSS) aimed at designing and deploying industrial smart plug pipelines to solve forecasting problems. The novel contributions of this research include:

- Extending the conventional DSS with the Explorative Management Subsystem, offering project-specific insights and recommendations.
- Integrating a Data Quality Module (DQM) to assess the quality of user inputs, ensuring reliable AI pipeline design.
- Maintaining a human-in-the-loop approach, giving users maximal control over every feature in the EX-DSS.
- Providing a practical demonstration of the EX-DSS through a software prototype for smart plug forecasting, validating the system's capabilities and showcasing its effectiveness in real-world applications, thereby highlighting its potential impact on reducing energy consumption in commercial buildings.

This chapter is structured as follows. Section 4.2 presents related works. Section 4.3 describes the challenges and contributions. In Section 4.4, the EX-DSS architecture framework is introduced. Section 4.5 describes the application of this framework in the design of smart plug forecasting AI pipelines via a software prototype. In Section 4.6, results are discussed, and finally, in Section 4.7, conclusions are drawn, and future research lines are presented.

4.2 Background

A Decision Support System (DSS) is software that helps users analyze data and make decisions. It generates insights and suggestions through a structured framework [62]. The classical DSS includes a **Data Management Subsystem**, which stores and handles data; a **Model Management Subsystem**, which manages models for DSS tasks; a **Knowledge Management Subsystem**, which provides information to users; and a **User Interface**, which connects users with DSS subsystems [105].

DSS supports semi-structured or unstructured decisions, requiring evaluations beyond mathematical modeling [27]. For example, buying a new tool [48] or deciding on stock market entry/exit [16]. These systems use rule-based models and AI to predict scenarios and derive insights [71].

DSS applications are used in various fields like business, healthcare, and agriculture. They can help medical staff plan and monitor medications [92] and manage production costs in agriculture [81]. Tools like Rapidminer[1] and Knime[2] support the design of AI pipelines but lack assistance during the configuration phases.

The problem considered in the EX-DSS software prototype involves forecasting the load of electrical appliances, monitored and controlled by smart plugs to optimize energy consumption [20]. Smart plugs can significantly reduce energy usage in commercial buildings by scheduling appliance operations. Accurate load forecasting is crucial for creating effective schedules. Current methods, including time series analysis and AI-based techniques like neural networks and ensemble methods [8], are time-consuming and complex to implement. This underscores the need for a robust DSS framework like EX-DSS to streamline the process and improve accuracy.

This chapter introduces a knowledge-driven DSS framework that incorporates detailed domain knowledge. This makes it especially useful for complex decision-making processes that require specific expertise. Specifically tailored for time series forecasting, this framework focuses on input quality assessment and fosters a collaborative research community by enabling users to share insights, models, and results.

[1] `https://altair.com/altair-rapidminer`
[2] `https://www.knime.com/`

4.3 Chapter Objectives and Challenges

- **Challenges in AI Pipelines for Smart Plug Forecasting**
  - Deploying AI pipelines for energy forecasting involves challenges related to complexity, Data Quality (DQ), and accessibility.
  - While AI methods such as time series analysis and deep learning are effective for predicting appliance loads, configuring these pipelines requires expertise in machine learning and domain knowledge.
  - The complexity of AI models can make them difficult to use for individuals without extensive technical backgrounds, limiting widespread adoption in energy optimization for commercial buildings.
- **Challenge 1: Complexity in AI Pipeline Configuration**
  - Users often struggle with selecting appropriate methods, tools, and hyperparameters for their specific forecasting needs.
  - Managing data preprocessing, feature selection, and model tuning requires a structured approach to prevent errors and inefficiencies.
  - Without a clear framework, these challenges can lead to inaccurate predictions and hinder successful AI implementation.
- **Challenge 2: Impact of Data Quality on Forecast Accuracy**
  - Poor DQ negatively affects AI forecasts, reducing the effectiveness of energy-saving strategies.
  - Many existing systems provide only basic data management tools, without integrated solutions for assessing and improving input DQ.
  - Ensuring high-quality data inputs is essential for producing reliable and interpretable AI models.
- **Proposed Solution: Explorative Decision Support System (EX-DSS)**
  - EX-DSS is a framework designed to simplify the design, configuration, and deployment of AI pipelines for smart plug forecasting.
  - It includes a **Data Quality Module (DQM)** that allows users to assess and enhance DQ at multiple stages of the pipeline.
  - By improving input DQ, EX-DSS helps ensure models are built on reliable information.
- **Human-in-the-Loop Approach for Better Decision-Making**

  - EX-DSS empowers users by combining expert knowledge with system-generated recommendations.
  - This approach enhances model interpretability and ensures users retain control over the AI pipeline design process.
  - By offering guided interactions, EX-DSS is designed to make advanced AI tools accessible to users of varying expertise levels, although this was not formally evaluated.

- **Collaboration and Knowledge Sharing**
  - EX-DSS encourages users to share models, insights, and results to foster collaboration within the research community.
  - A shared knowledge base can lead to improvements in forecasting techniques and support the adoption of AI-driven energy management strategies.

- **Software Prototype and Real-World Application**
  - A software prototype was developed to demonstrate how EX-DSS streamlines AI pipeline design for smart plug forecasting.
  - The system showcases how AI can optimize energy consumption in commercial buildings, contributing to sustainable energy use.

- **Overall Impact of EX-DSS**
  - Reduces the complexity of AI pipeline deployment for energy forecasting applications.
  - Addresses DQ challenges by integrating assessment and improvement tools.
  - Enhances accessibility, making AI-based energy management solutions available to a broader audience.
  - Paves the way for more efficient and scalable AI-driven energy optimization strategies.

4.4 EX-DSS framework

As mentioned in Section 4.2, a traditional DSS has three subsystems and a User Interface (UI). The new EX-DSS adds a DQM (Fig.4.1C) and an Explorative Management Subsystem (Fig.4.1D.4). DQM (Fig.4.1C) improves DQ and promotes information sharing. The Explorative Management Subsystem (Fig.4.1D.4) provides project-specific insights for better analysis.

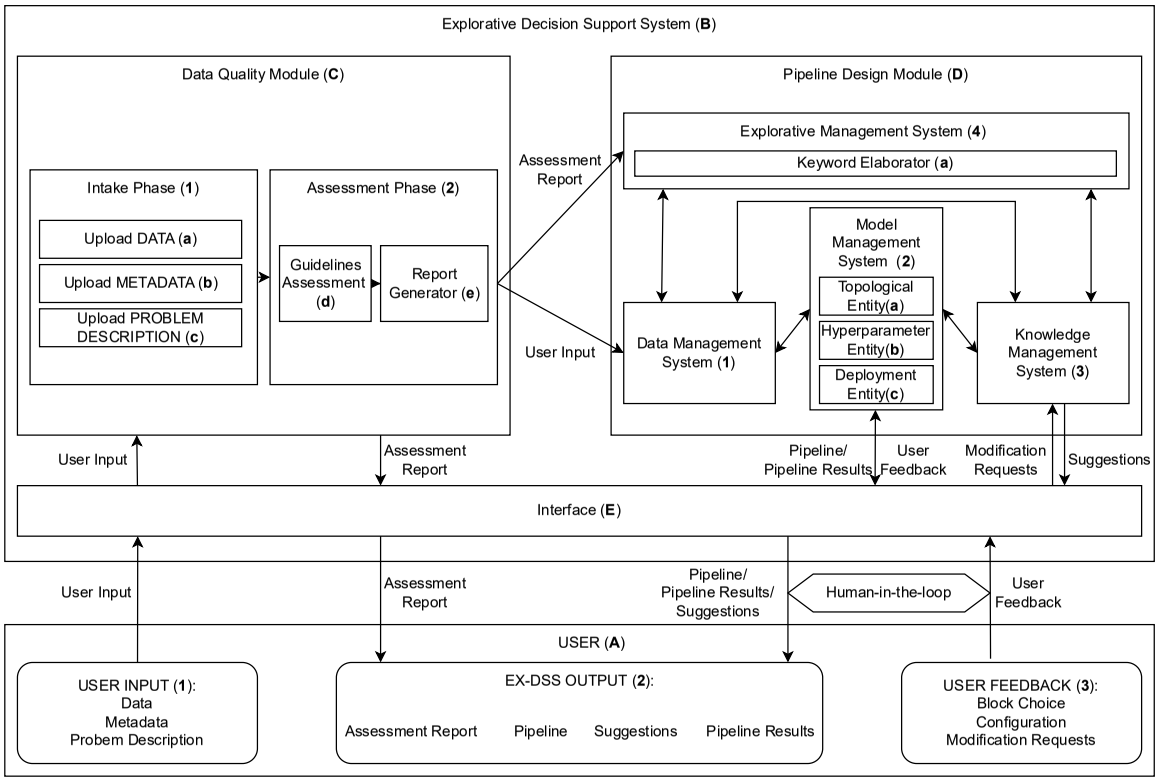


Figure 4.1. **EX-DSS architecture framework.** This illustrates the proposed EX-DSS framework. (A) User-level input and outputs. (B) Blocks composing the EX-DSS: (C) Block to assess the quality of the user's input and (D) Block to aid the users in designing, configuring, and deploying the pipeline. (E) User Interface that allows the user to dialogue with the EX-DSS. The arrows represent the flow of information inside the systems.

The EX-DSS architecture framework has two levels: the User (Fig.4.1A) and the EX-DSS (Fig.4.1B). The User interacts with the system to solve problems using semi-automated steps. The EX-DSS analyzes problems, provides extra information, and helps create the AI pipeline. A Graphical User Interface (Fig.4.1E) facilitates these interactions.

The EX-DSS has two main parts: DQM (Fig.4.1C) and the Pipeline Design Module (Fig.4.1D). DQM includes the Intake Phase (Fig.4.1C.1) for uploading data and the Assessment Phase (Fig.4.1C.2) for evaluating DQ by generating a report. The Pipeline Design Module (Fig.4.1D) has four subsystems: Data Management (Fig.4.1D.1), which stores essential information; Model Management (Fig.4.1D.2), which manages the pipeline structure (Fig.4.1D.2.a), configuration (Fig.4.1D.2.b) and training (Fig.4.1D.2.c); Knowledge Management (Fig.4.1D.3), which provides general information; and Explorative Management (Fig.4.1D.4), which derives project-specific insights from the keywords extracted from the DQ report (Fig.4.1D.4.a).

The general knowledge in the Knowledge Management Subsystem (Fig.4.1D.3) is not tied to specific problems. For example, it answers questions like "What is forecasting?". Project-specific knowledge in the Explorative Management Subsystem (Fig.4.1D.4) is detailed and tailored to specific projects, like "What are the best methods for forecasting smart plugs?". The DSS is called "explorative" because it helps users dive into and explore specific problems, such as forecasting smart plugs for energy savings in buildings.

By adding an extra subsystem, the EX-DSS improves modularity, maintainability, and reusability. It separates general knowledge from project-specific information, allowing better customization and system performance.

4.5 An industrial application of the framework as software prototype: EX-DSS for Smart Plug Forecasting Pipelines

This section presents an industrial application of the Explorative Decision Support System (EX-DSS) framework, demonstrating its implementation through a software prototype developed for smart plug forecasting pipelines. The EX-DSS prototype provides a structured and guided environment for designing, configuring, and analyzing AI forecasting pipelines, helping users make data-driven decisions to optimize energy consumption. The system incorporates expert-driven insights and Natural Language Processing (NLP)-based recommendations to assist users in selecting and fine-tuning AI models.

The prototype is built to support energy analysts and researchers in evaluating multiple forecasting methodologies and selecting the most effective AI pipeline for smart plug scheduling. Users can upload and preprocess datasets, configure pipelines through a user-friendly interface designed to be intuitive, execute models on an external computing service, and compare results using key performance metrics. The EX-DSS framework streamlines the exploration of different AI models, aiming to make forecasting pipelines more interpretable and adaptable, based on design intentions and informal feedback to various energy management scenarios.

This industrial application serves as a prototype, showcasing how EX-DSS can enhance AI-driven decision-making in energy forecasting. The following sections provide a detailed overview of the software prototype's architecture, functionalities, and real-life application using a smart plug dataset. This case study demonstrates the framework's ability to support interactive AI pipeline design, model evaluation, and energy optimization, validating its potential for future real-world deployment in energy management systems.

4.5.1 General Description

- **Software Demo Title**: EX-DSS for Smart Plug Forecasting Pipelines
- **Overview**: The EX-DSS is developed to assist users in designing, configuring, and analyzing AI pipelines for smart plug forecasting. The software prototype enables users to configure AI pipelines with expert-driven and Natural Language Processing (NLP)-supported recommendations, optimizing energy consumption by forecasting plug usage and scheduling appliances efficiently.
- **Objectives**:
    - Provide an interactive framework for configuring and understanding AI pipelines.
    - Enhance usability through expert suggestions and NLP-driven insights.
    - Assist in forecasting energy consumption for smart plugs.
    - Enable users to compare and evaluate multiple pipeline configurations.
- **Scope**:
    - **Targeted Functionalities**:

  - **Pipeline Configuration**: Users can build and modify AI pipelines for smart plug forecasting.
  - **Expert-Driven Assistance**: NLP integration provides real-time recommendations.
  - **Data Management**: Users can upload and preprocess smart plug datasets.
  - **Pipeline Execution**: Configured AI pipelines are deployed and executed via an external container.
  - **Comparison Feature**: Users can compare the results of different pipeline configurations.
- **Excluded Functionalities**:
  - Large-scale deployment features (planned for future work).
  - Integration with domains beyond smart plug forecasting.
  - Advanced real-time data processing and adaptive learning mechanisms.

- **System's Requirements**
  - **Functional Requirements**:
    - Dataset management (upload, description, metadata handling).
    - Project initialization with relevant metadata.
    - AI pipeline configuration and visualization.
    - Training and execution monitoring.
    - Result analysis and evaluation with visual insights.
  - **Non-Functional Requirements**:
    - Efficient system performance for data processing.
    - Secure storage of user data and project configurations.
    - Intuitive user interface for ease of use.
    - Scalability to accommodate various dataset sizes.
    - Compatibility with common web browsers.

- **Architecture**
  - **Software Network-based Architecture**: EX-DSS follows a client-server model. Users interact with the DSS through a web-based interface, while an external containerized service handles pipeline execution. The DSS web server processes user inputs, retrieves expert insights, and sends pipeline configurations to the external server for execution. The final outputs include structured predictions, performance reports, and AI recommendations.

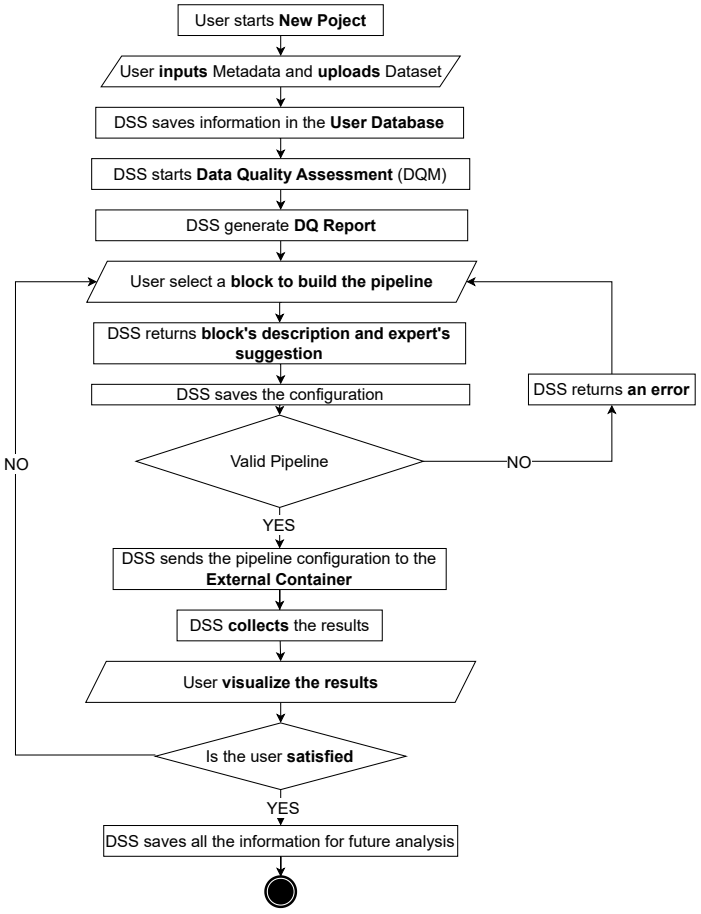


Figure 4.2. **Processing Flow of the EX-DSS Framework.** The diagram represents the structured interaction between users, the DSS web server, and the AI pipeline execution. Users iteratively refine and explore different AI configurations before finalizing results.

- **Processing Flow**: The EX-DSS workflow is structured into three key phases, facilitating seamless interaction between users, the DSS system, and the AI pipeline execution. Figure 4.2 shows the process flow of the user interacting with the EX-DSS prototype.

  1. **Project Initialization Phase**:
     - Users create a new project and provide metadata.
     - The system allows dataset uploads and metadata entry.
     - Keywords are selected to improve pipeline recommendations.
     - A Data Quality assessment is performed to check completeness, consistency, and timeliness.
     - A structured Data Quality (DQ) report is generated.
  2. **Pipeline Configuration Phase**:
     - Users design and modify AI pipelines using an interactive interface.
     - The system retrieves expert recommendations based on dataset keywords.
     - AI-driven insights assist users in configuring pipeline blocks.
     - The system ensures the correct interconnection of pipeline components.
     - Users iteratively adjust configurations based on system feedback.
  3. **Execution and Analysis Phase**:
     - The configured pipeline is sent for execution in an external processing environment.
     - The system trains models using the uploaded dataset.
     - Users analyze forecasting results via performance metrics and visualizations.
     - The system supports pipeline comparison, enabling users to evaluate multiple configurations.
     - If the results are unsatisfactory, users can modify the parameters and re-run the pipeline.
     - The final trained model and predictions can be downloaded for further use.

- **Main User Stories**: Below, there are 2 user stories where target clients describe the actions that they need the software to perform.

  - **User Story 1**: *As a Data Energy Analyst, I want to configure a smart plug forecasting pipeline, so that I can optimize plug scheduling and reduce energy consumption.*

    - **Acceptance Criteria**:
        - Users can create and save a new project with metadata.
        - Users can upload and preprocess smart plug datasets.
        - The system provides expert guidance on forecasting methods.
        - The pipeline configuration ensures proper block connectivity.

- **User Story 2**: *As a Data Scientist, I want to quickly configure and test multiple forecasting pipelines without writing code, so that I can efficiently evaluate AI models for smart plug forecasting.*
    - **Acceptance Criteria**:
        - Users can set up AI pipelines through a no-code graphical interface.
        - The system provides pre-configured templates for standard forecasting models.
        - Users can automatically optimize pipeline parameters with minimal manual input.
        - Execution time is minimized by using parallel processing or efficient caching mechanisms.
        - The system generates automated insights comparing different pipelines.
        - Users can download the best-performing model and results for further analysis.

4.5.2 EX-DSS Real-Life implementation as software prototype

This subsection presents the use case and application context of the Explorative Decision Support System (EX-DSS) framework for the design of an AI-powered smart plug forecasting pipeline through a software prototype. The EX-DSS framework, introduced in Section 4.4, provides a structured approach to developing and configuring AI models for energy-efficient appliance management in commercial buildings.

Smart plugs, which act as intermediaries between appliances' power cords and wall sockets, transform standard devices into smart, controllable systems. These devices are widely used to monitor plug loads from appliances such as printers, copiers, and televisions. Research indicates that plug loads (excluding lighting, HVAC, and water heating) contribute to over 40% of total energy consumption in commercial buildings, with this figure continuing to rise over time [20, 106]. By leveraging automated control and predictive modeling, smart plugs can reduce energy waste by up to 20%, enhancing overall building energy efficiency.

Artificial Intelligence (AI) plays a critical role in identifying plug usage patterns, forecasting energy consumption, and scheduling devices to optimize electricity usage [8]. However, designing and selecting the most effective AI pipeline for smart plug forecasting can be time-consuming and complex. The software prototype developed in this chapter addresses this challenge by providing a guided environment for researchers to design, evaluate, and refine AI pipelines for smart plug forecasting and scheduling. By streamlining this process, the system facilitates energy-efficient decision-making while allowing trained models from the design phase to be downloaded for further application.

4.5.3 Software Prototype Description

The EX-DSS software prototype is a web application developed in Python using the Dash framework[3]. It assists users by providing descriptions and suggestions for configuring each block of the AI pipeline, based on issue-specific keywords related to the Smart Plug problem described in Section 4.5.2 and listed in Appendix C. In addition, a chatbox enables users to pose general questions. This component is integrated with a Natural Language Processing (NLP) model developed by Cohere[4].

The functionalities of the EX-DSS prototype are illustrated through an experiment that uses methods and datasets collected by Botman et al. (2024) [8]. While Botman et al. proposed multiple alternatives[5], a subset of these methods has been included in the current implementation. After pipeline configuration, EX-DSS connects to an external server for workflow execution via the Airflow platform[6].

A video demonstration of the EX-DSS prototype is available at the following online resource: EX-DSS video demonstration[7].

New Dataset

The EX-DSS software prototype's experiment began with uploading a new dataset (Fig.4.1C.1). The user provided a title, data source, and a short description (Fig.4.1C.1.b) to facilitate understanding and improve support in subsequent phases. Next, the user selected keywords from a set proposed by

[3] https://dash.plotly.com/

[4] https://cohere.com/

[5] https://gitlab.esat.kuleuven.be/2.Botman/smart-plug-pipeline

[6] https://airflow.apache.org/

[7] https://drive.google.com/drive/folders/1gfoEVD-IIzN0zArnx0bMuGNdSMxARamP?usp=sharing

Timestamp	**Plug 0**	**Plug 1**	**Plug 2**	**...**	**Plug 167**	**Plug 168**
2021-11-18 11:45:00	NaN	NaN	NaN	...	NaN	NaN
...	...	...	...	...	...	...
2021-11-18 13:15:00	NaN	10.413	19.656	...	NaN	NaN
...	...	...	...	...	...	...
2023-03-31 17:00:00	55.809	10.03	19.44	...	2.64	4.176

Table 4.1. **Snapshot of the Smart Plugs Preprocessed Dataset.** The dataset consists of power consumption data from 169 smart plugs, each monitoring various electrical appliances. Power values are recorded in milliwatts (mW) at fifteen-minute intervals, spanning from November 18th, 2021, to March 31st, 2023. The table displays a subset of the data, including time series for 146 printers, 16 copiers, 4 TVs, and 3 fax machines.

the EX-DSS (see Appendix C for the full list). These keywords are crucial for the Explorative Management Subsystem to generate insights and target specific pipeline characteristics. Finally, the user submitted the main data file (Fig.4.1C.1.a) and could also upload supplementary files such as metadata or additional information (Fig.4.1C.1.b).

For this experiment, the input was based on the work proposed by Botman et al. (2024) [8]:

- **Title**: Smart Plug Data,
- **Data source**: "`https://gitlab.esat.kuleuven.be/2.Botman/smart-plug-pipeline/-/tree/main/Dataset`"
- **Data description**: "The dataset is collected through smart plug sockets between the wall plugs and the electric appliance as detailed in Chia et al. (2023) [20]. Smart plugs from Best Energy Reduction Technologies (BERT) are deployed in fifteen buildings on the campus of the University of California, San Diego (UCSD). The power level of each appliance is recorded in mW at fifteen-minute intervals. The dataset consists of 169 high-quality smart plug time series spanning 498 days, from November 18th, 2021, to March 31st, 2023. The 169 plug loads include 146 printers, 16 copiers, 4 TVs, and 3 fax machines. This dataset is openly accessible; see Botman et al. (2024) [8] for more details."
- **Keywords selected**: "Smart Plugs, High Performance, Non-intrusive"
- **Main file**: "SmartPlugPreprocessedData.csv"

- **Additional metadata files**: ["SmartPlugMetaData.csv" "SmartPlugHolidayData.csv"]

Table 4.1 shows a snapshot of the main dataset, "SmartPlugPreprocessedData.csv." Rows indicate the time at which power values are recorded, and columns represent monitored electrical appliances with power values in mW. The goal is to predict these power values. In the illustrated subset, recordings start at 11:45 on the 18th of November 2021 until 17:00 on the 31st, 2023, capturing the power load of 169 smart plugs at fifteen-minute intervals.

Assessment Report

The Assessment Phase (Fig.4.1C.2) begins once the system receives the data and the complementary input. In this phase, DQM in EX-DSS runs an internal analysis to assess the quality of the inputs used in subsequent steps to generate suggestions based on the work by Rinaldi et al. (2023) [78].

Although the Assessment Phase can be set up to run different guidelines, for the software prototype, three guidelines were chosen (Fig.4.1C.2.d):

- **FAIR Paradigms** [110]: The system checks dataset uniqueness and corrects saving (findability), attempts to open the dataset in a pandas dataframe[8] (reusability), evaluates data encoding (interoperability), and audited security rules (accessibility).
- **DQ Analysis**: This included checking for missing values (completeness), assessing value types (consistency), measuring distance between input and reference datasets (accuracy), and verifying if the data is up-to-date (timeliness).
- **DC Automation**: This involved routines like evaluating missing values, identifying columns with a single value, and checking for row duplication.

The clarity of the dataset description is also evaluated. At the end of this phase, the system provides two scores: one for DQ analysis and another for DC assessment, as shown in Rinaldi et al. (2023) [78].

Although this analysis is done in the background, users could configure it manually. They can choose whether to analyze only the main file or include metadata files, select the weight of each quality metric, and decide if any analysis

[8] `https://pandas.pydata.org/`

should be skipped. All this information is compiled into a report (Fig.4.1C.2.b), which users can download.

Listing 4.1 shows a summary of the report that contains the analysis with the above-mentioned guidelines.

Listing 4.1. **Sample JSON Report Extract**: This excerpt presents the findings from the EX-DSS Data Quality and Cleaning assessment. The report includes scores for data quality and cleaning, along with detailed analysis for each guideline, such as FAIR principles, timeliness, reusability, and data description clarity. The report highlights key aspects such as dataset completeness, consistency, and the detection of data anomalies like missing values or duplicate rows.

```
{"SCORES" {
    "Data Quality": 0.56,
    "Data Cleaning": 0.99},
"FAIR" {
    "Reusability": "The dataset was rightly converted to
        a dataframe."
    "Data_description_clarity_score": "Input text clarity
        score: 85%", ...},
"DATA QUALITY" {
    "Timeliness": "The dataset is not up to date ->
        parameter is 0. The user set up the threshold to 0
        years." ...},
"DATA CLEANING" {
    "Time Column": "Identified."
    "Single Columns": "There are no columns with only one
        value." ...}
```

New Project

The user needed to define the project's title and description, upload any additional documents that could help understand the project's final goal, and select the most pertinent keywords from a subset proposed by the EX-DSS (see Appendix C for the full keywords list). These keywords assist in formulating proper suggestions and clarifying the research intention. An example of a new project record is as follows:

- **Title**: Smart Plug Project,
- **Project's description**: "This project implements smart plug active operating mode detection, plug-level load forecasting, and a plug scheduling methodology. A pipeline integrates the detected operating

modes with forecasting and scheduling, aiming at reducing building energy consumption [8]."

- **Additional documents**: "No additional description documents"
- **Keywords selected**: "High Performance, Non-intrusive"

Pipeline Configuration and the EX-DSS Guidance

When a project is initialized, the user can start the design and configuration of the pipeline. Figures 4.3, 4.4, 4.5, and 4.6 show screenshots of the configuration page of the EX-DSS prototype software. Figure 4.3 lists the available blocks developed in the EX-DSS software prototype, while Figure 4.4 displays the suggestions generated by the Knowledge Management Subsystem with support from the Explorative Management Subsystem.

A suggestion consists of a block's description, insight generated by the Explorative Management for the specific project, and a chatbox for communication with the Knowledge Management Subsystem. The Explorative Management Subsystem (Fig.4.1D.4) uses the project's keywords to find the closest matching dataset and generate suggestions based on stored expert knowledge. These suggestions include a list of blocks and methods, such as the appropriate forecasting method or technique to preprocess the dataset. This modular approach provides accurate and specialized knowledge, which can be modified to enhance specific subject knowledge or reused for other purposes.

The Knowledge Management Subsystem (Fig.4.1D.3) maintains a general knowledge view. In the EX-DSS software prototype, it uses a Generative AI model developed by Cohere[9]. The system connects to the Cohere platform via an API, sends a request with a question, and displays the AI-generated response to the user.

Additionally, the Knowledge Management Subsystem ensures that the interconnections between blocks are correct. For example, in the EX-DSS software prototype, the "Forecast" block could not be followed by the "Dataset" block. Figure 4.5 represents the configuration area, where users choose methods and configure parameters. For example, in the "Forecast" block, users can select the forecasting method and the prediction horizon. In the example shown (Fig. 4.5), "Global XGBoost" was chosen with a prediction horizon of "1" day.

Figure 4.6 shows the area where the user can organize and connect the chosen blocks. In the example, the tasks were organized as follows: "Dataset"

[9] `https://cohere.com/`

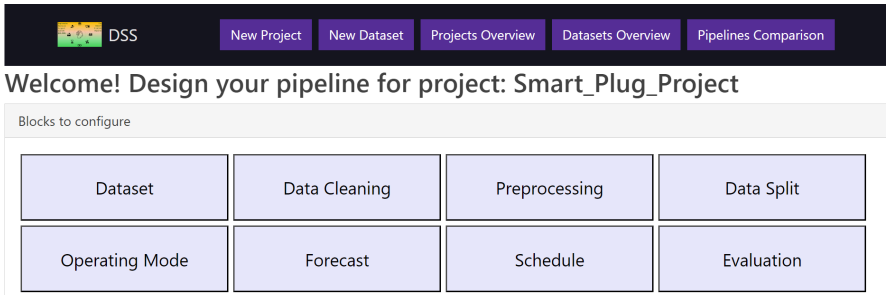


Figure 4.3. **EX-DSS Configuration Page - List of Available Blocks.** This screenshot showcases the blocks developed within the EX-DSS software prototype. Users can choose from these blocks to design their data processing and forecasting pipeline.

for selecting the dataset, "Operating Mode" for determining the appliance's operating mode, "Forecast" for predicting the appliance loads, "Schedule" for planning when the appliances should be turned on or off, and "Evaluate" for assessing the performance of the other blocks. "Data Cleaning" and "Preprocessing" were excluded since the dataset was already preprocessed.

Once the configuration for each block is saved, the pipeline is ready for training.

Pipeline Training and Results

Once the pipeline is ready, the user initiates the Model Management Subsystem (Fig.4.1B.2), specifically the Deployment Entity (Fig.4.1B.2.C), to launch the deployment process. In the EX-DSS software prototype, this was implemented by an external server running Airflow[10], a platform designed for executing and monitoring a chain of tasks. The Deployment Entity transferred the necessary files to the Airflow server. Figure 4.7 displays the pipeline's runtime, which took 9 minutes and 24 seconds to train and generate results.

Upon completion, the Airflow server communicates the results back to the EX-DSS, and users can download these results through the EX-DSS interface. Additionally, users can download the forecasting model, operating mode model, predictions, and final schedule. Table 4.2 provides an example of the final schedule. Additionally, Table 4.3 displays the numerical values of key metrics used to assess smart plug scheduling performance after applying the "Global XGBoost" method. The number of violations quantifies how often appliances were mistakenly turned off while in use, while missed chances reflect instances where energy could have been saved but wasn't. Energy saved represents

[10] `https://airflow.apache.org/`

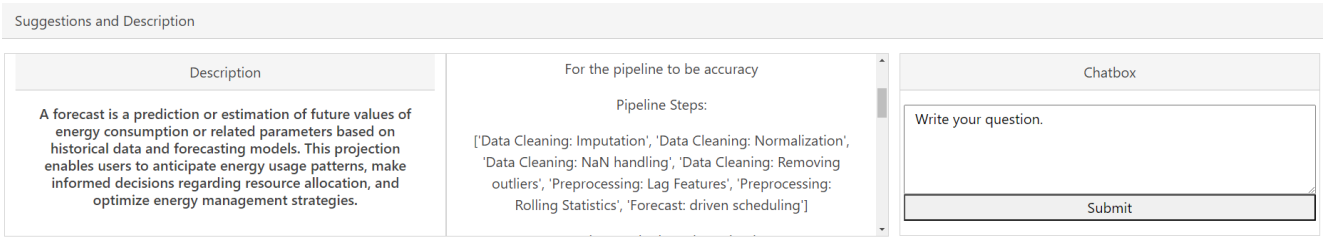


Figure 4.4. **EX-DSS Configuration Page - Block Suggestions.** The system provides users with tailored suggestions for each block, which include a description, expert insights based on the project's keywords, and a communication interface with the Knowledge Management Subsystem. These suggestions guide users in selecting appropriate methods and techniques for their dataset.

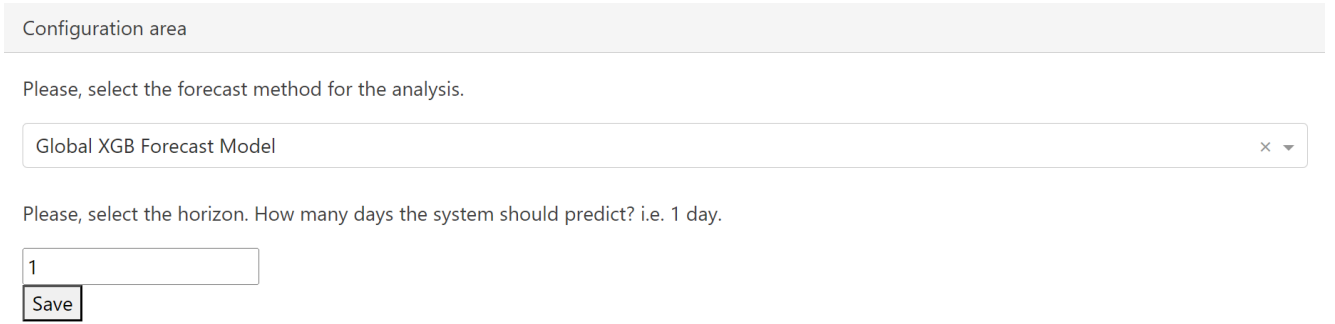


Figure 4.5. **EX-DSS Configuration Page - Block Configuration Area.** This section allows users to configure parameters for each selected block. In the shown example, the user configures the "Forecast" block by selecting the forecasting method, "Global XGBoost," and setting the prediction horizon to one day.

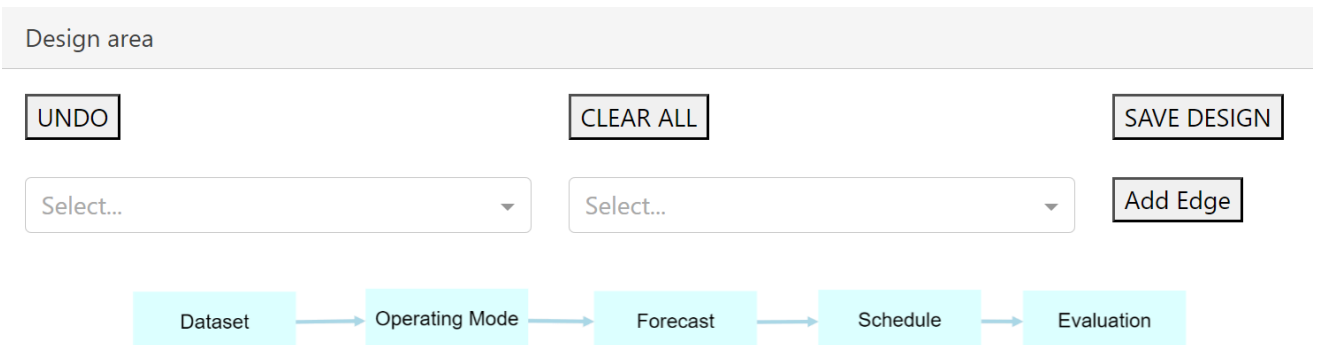


Figure 4.6. **EX-DSS Configuration Page - Design Area.** The design area allows users to organize and connect blocks to define the data processing and forecasting pipeline. In this example, the blocks for dataset selection, appliance operating mode, forecasting, scheduling, and evaluation are connected, while "Data Cleaning" and "Preprocessing" are omitted as the dataset is already preprocessed.

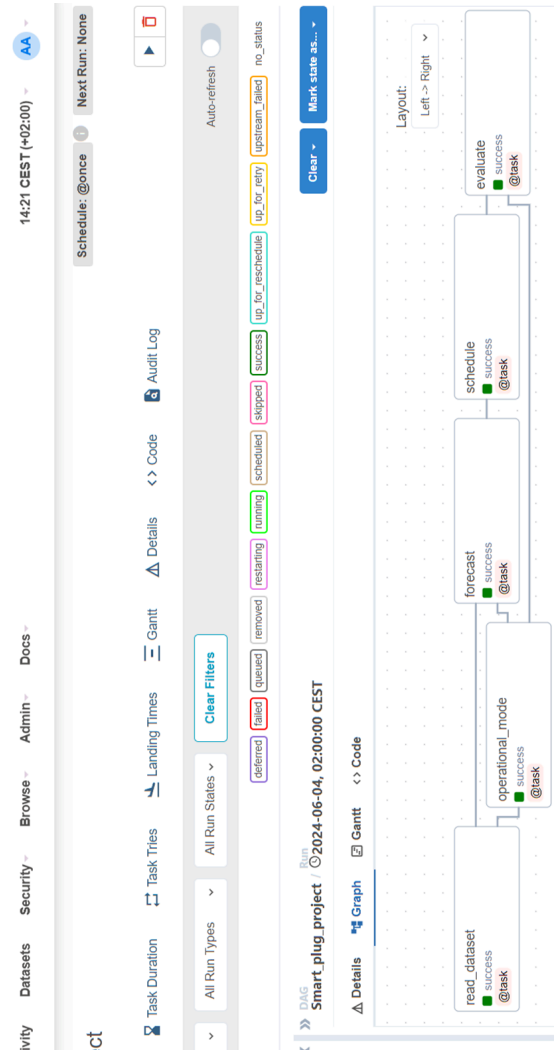


Figure 4.7. **Airflow Server Runtime Screenshot.** This screenshot displays the Airflow service used by the EX-DSS software prototype to execute the pipeline. It shows the runtime details of the training process, which took 9 minutes and 24 seconds to complete.

the percentage reduction in consumption compared to no scheduling, and the number of turn on/off commands per plug per day gives an idea of how often devices are switched, which can impact appliance wear. Lastly, energy efficiency measures the proportion of energy used when appliances are actually active. More technical details and context can be found in Botman et al. (2024) [8].

Listing 4.2. **Static Suggestions Stored in the EX-DSS Software Prototype.** This example shows the structure of static suggestions stored within the EX-DSS, which include the steps for data cleaning, operating mode detection, and forecasting methods like "Global XGBoost" and "Global FNN.".

```
{"Pipeline Steps" {
    "Data Cleaning" "Imputation"
    "Data Cleaning" "Normalization" ...},
"Operating Mode Detection" {
    "Method" "GMM"
    "Method" "Ensemble"
"Forecasting" {
    "Method" "Global XGBoost"
    "Method" "Global FNN"}
```

4.5.4 Implementation Details

- **Technology Stack** The following technologies were used in the development of the EX-DSS software prototype.
    - **Frontend**: Dash[11] – A Python-based framework for building interactive web applications and data visualizations.
    - **Backend**: Flask[12] – A lightweight Python micro-framework for handling server-side logic, user requests, and API interactions.
    - **Database**: Json File System – A relational database to store user configurations, metadata, and model results efficiently.
    - **Workflow Management**: Apache Airflow[13] – Manages and schedules AI pipeline execution on an external server, ensuring modular and structured workflows.
    - **Machine Learning Frameworks**:
        - Scikit-learn (Sklearn)[14] – Provides utilities for data preprocessing, feature selection, and classical forecasting models.

[11] https://dash.plotly.com/
[12] https://flask.palletsprojects.com/en/stable/
[13] https://airflow.apache.org/
[14] https://scikit-learn.org/

Timestamp	Plug_0	...	Plug _168
2023-01-12 00:00:00	OFF	...	OFF
2023-01-13 00:15:00	OFF	...	OFF
...	...	...	...
2023-01-13 05:00:00	OFF	...	OFF
2023-01-13 05:15:00	ON	...	OFF
...	...	...	...
2023-01-13 09:15:00	ON	...	OFF
2023-01-13 09:30:00	ON	...	ON
...	...	...	...
2023-01-13 12:15:00	ON	...	ON
2023-01-13 12:30:00	OFF	...	ON
...	...	...	...
2023-01-13 16:30:00	OFF	...	ON
2023-01-13 16:45:00	OFF	...	OFF
...	...	...	...

Table 4.2. **Schedule Overview.** The table provides an example of the schedule for plugs 0 and 168. Both plugs are off at night and turn on in the morning, with plug 0 turning on earlier. Plug 0 turns off around lunch while plug 168 turns off later in the afternoon. The schedule can be summarized with the turn-on and turn-off times: Plug 0: {2023-01-13 05:15:00: "Turn on"; 2023-01-13 12:30:00: "Turn off"}, Plug 168: {2023-01-13 09:30:00: "Turn on"; 2023-01-13 16:45:00: "Turn off"}

	Global XGBoost Schedule
Number of violations (%)	3.27
Missed chances (%)	27.67
Energy saved (%)	25.84
Number of turn on/off commands per plug per day	3.33
Energy Efficiency (%)	47.92

Table 4.3. **Forecasting Evaluation Metrics.** This table summarizes the performance of the "Global XGBoost" forecasting method in the pipeline, including key metrics such as the number of violations, missed chances, energy saved, and energy efficiency.

    - TensorFlow[15] – Used for deep learning models within the smart plug forecasting pipeline.
    - Cohere API[16] – Integrated Generative AI model for Natural Language Processing (NLP)-driven guidance, helping users understand pipeline configurations.
  - **Containerization & Deployment**: Docker[17] – Ensures an isolated execution environment, allowing reproducibility and consistency of AI pipelines.
  - **Version Control & AI Pipeline Repository**: GitLab Repository[18] – The core AI pipeline implementation is based on pre-existing methods for smart plug forecasting design by Botman et al. (2024) [8].
  - **Visualization Libraries**: Matplotlib and Plotly – Used for graphical representations of pipeline structures, results, and forecasting comparisons.
  - **Library Dependencies**: A complete list of libraries used in the EX-DSS software prototype can be found in Appendix B.

- **Test Cases Performed** The EX-DSS prototype was extensively tested to ensure usability, robustness, and performance. The following test cases were executed:

  - **Project Initialization**: Verified that users could create, save, and retrieve new projects. Ensured correct storage of metadata and configuration settings.
  - **Dataset Management**: Confirmed that users could upload and preprocess datasets correctly. Assessed metadata handling and dataset description storage.
  - **Pipeline Configuration**: Validated that users could modify, enable, and adjust AI pipeline components. Tested NLP-assisted recommendations for block configurations.
  - **Pipeline Visualization**: Ensured interactive representation of pipeline structures. Verified correct integration of expert-driven insights into the interface.
  - **Pipeline Execution & Training**: Tested pipeline execution using Apache Airflow and Docker-based training servers. Verified successful model training, inference, and result generation.

[15] `https://www.tensorflow.org/`
[16] `https://cohere.com/`
[17] `https://www.docker.com/`
[18] `https://gitlab.esat.kuleuven.be/2.Botman/smart-plug-pipeline`

- **Result Analysis & Evaluation**: Checked the accuracy and consistency of forecasting results. Assessed the effectiveness of comparison features between pipeline variations.
- **Usability & Accessibility**: No formal usability study was conducted; all feedback was gathered informally during development and internal testing. The prototype was developed for research purposes only and not for direct end-user deployment.
- **Error Handling & System Robustness**: Simulated user errors (e.g., invalid configurations, missing data) to assess system responses. Implemented error messages and fallback mechanisms for unexpected input.

- **Limitations and Constraints** Despite its capabilities, the current version of EX-DSS has some limitations and known issues, as outlined below.

  - **Incomplete Features**
    - Automated hyperparameter tuning is not yet implemented (planned for future updates).
    - Support for additional AI domains beyond smart plug forecasting is still under development.
    - Limited explainability features.
    - Multi-user collaboration features are not yet available—only single-user interaction is supported.
  - **Known Issues**
    - Manual configuration of hyperparameters is required, which may be challenging for non-expert users.
    - Some pipeline components have not been optimized for large-scale datasets.
    - Initial loading times for large datasets may be slow due to preprocessing overhead.
    - Execution errors can occur in complex configurations, sometimes without clear error messages.
    - The current system is designed for batch processing only, with limited real-time data streaming support.
    - Cloud deployment requires additional configuration steps to ensure compatibility with different infrastructures.

4.6 Results and Discussion

The study's findings indicate that the Explorative Decision Support System (EX-DSS) architecture framework enhances the design and implementation of DSS for designing smart plug pipelines, optimizing data forecasting, and, consequently, aiding in reducing energy consumption. By extending the classical Decision Support System (DSS) framework, the EX-DSS incorporates a module for assessing user input quality, facilitating collaboration among users, and promoting reproducibility. This prevents the repetition of previous errors and accelerates the pipeline creation process.

The EX-DSS allows users to upload new datasets, along with descriptions and additional documents, providing a comprehensive overview. The system internally evaluates these inputs and generates a report on DQ and description clarity. By maintaining the essential human-in-the-loop characteristic of a DSS, users retain control over the types of analyses the system should perform, ensuring that the information received is of high quality and increasing the likelihood of high-performance outcomes from the deployed pipeline.

A key innovation in the EX-DSS is the inclusion of the Explorative Management Subsystem, which supports the Knowledge Management Subsystem by analyzing user inputs and providing relevant insights. By using a system of keywords tailored to the smart plug forecasting problem, the EX-DSS offers specialized support while maintaining general intelligence within the Knowledge Management Subsystem, thanks to the integration of the Cohere generative AI model. This dual support (general and specific) significantly aids users in designing and configuring forecasting pipelines, allowing for customization and catering to various needs and requirements.

The study has shown that the pipeline design, configuration, and deployment process can be carried out without coding, making it accessible to a broader range of users. However, future work should include extensive testing with diverse datasets to further validate the system's robustness and flexibility. Additionally, user feedback should be collected to refine the interface and improve the overall user experience. Integrating real-time data processing and adaptive learning capabilities could also enhance the EX-DSS, making it more responsive to changing conditions and user needs.

4.7 Conclusions

The Explorative Decision Support System (EX-DSS) framework, implemented through a software prototype, has demonstrated its potential in streamlining the design of AI pipelines for smart plug scheduling, ultimately contributing to reducing energy consumption in commercial buildings. By extending the classical Decision Support System (DSS) architecture, EX-DSS introduces the Explorative Management Subsystem, which provides project-specific insights and recommendations. This structured approach ensures that users benefit from both generalized knowledge via the Knowledge Management Subsystem and tailored guidance through the Explorative Management Subsystem. This decoupling enhances adaptability and usability, making the system potentially more effective for users with different levels of expertise; however, this aspect has not yet been scientifically assessed.

A key enhancement within the EX-DSS is the integration of a Data Quality Module (DQM), which plays a crucial role in assessing and improving the reliability of input data. Given the well-documented impact of Data Quality (DQ) on AI model performance, this module allows users to evaluate dataset integrity, detect inconsistencies, and refine preprocessing strategies. By ensuring that AI pipelines are built upon high-quality data, EX-DSS enhances forecast accuracy and overall system robustness.

Another notable feature of EX-DSS is its collaborative and knowledge-sharing capabilities. The system promotes the reproducibility and shareability of pipeline configurations and dataset insights by requiring users to document their project rationales, methodological choices, and pipeline structures. This transparency fosters a collective learning process, helping users avoid redundant efforts, standardize best practices, and enhance the overall efficiency of AI pipeline development. By enabling users to compare and refine pipeline strategies, EX-DSS establishes a foundation for continuous improvement in forecasting methodologies.

Despite its promising contributions, this study acknowledges certain limitations. Firstly, the software is not production-ready, as it was designed primarily as a demonstration prototype. Secondly, the current implementation is limited to methods proposed by Botman et al. (2024) [8], restricting the system's ability to incorporate alternative forecasting techniques. Additionally, the development process required domain-specific expertise, yet due to time constraints and the limited availability of subject-matter experts, this study was primarily shaped by input from a single expert. Future work should address these challenges by expanding the system's methodological scope, increasing expert collaboration, and further refining the user experience.

In the future, several avenues for enhancement can be explored. An important research direction is further investigating the interaction between the Knowledge Management Subsystem and the Explorative Management Subsystem, refining how knowledge is structured and utilized across projects. Another potential improvement is the expansion of the software's visualization capabilities, allowing users to analyze results directly within the EX-DSS interface. Moreover, a comparative analysis feature could be developed, enabling users to evaluate multiple pipeline configurations and identify the most effective ones based on performance metrics. Finally, to broaden the accessibility of AI-driven energy forecasting, future developments should aim to adapt EX-DSS for large-scale deployment, integrating it with existing AI and decision-making platforms.

By addressing these areas, EX-DSS can evolve into a comprehensive, scalable, and user-friendly tool, further solidifying its role in advancing AI-based decision support systems for energy-efficient smart plug management and beyond.

Chapter 5

Conclusions

The frameworks and methodologies introduced in this thesis lay the groundwork for a **new generation of Decision Support Systems (DSS)**—ones that are not only intelligent and automated but also **transparent, human-centered, and adaptable** across multiple domains. As AI continues to shape decision-making processes, ensuring that these systems remain **accessible, ethical, and effective** is crucial. The insights gained from this research represent a significant step toward that future, shaping how **AI-powered DSS** evolve in the coming years.

This chapter consolidates the key findings of the research presented in this thesis, providing a synthesis of the **scientific contributions, real-world implications, and future research directions**. It reflects on the objectives outlined in Chapter 1, demonstrating how they were achieved through the development of a **novel framework for DSS** that integrates **Artificial Intelligence (AI) pipelines**, with a particular focus on **energy time series applications**. This chapter also explores the broader practical impact of this research and suggests pathways for further advancements in the field.

The Chapter is structured as follows:

- The first section, **Scientific Conclusions**, expands on the implications of this research, outlining its contributions to both theoretical foundations and practical applications, particularly in energy time series forecasting. It discusses how the proposed frameworks successfully **enhance AI pipeline design and deployment**.
- The **Cross-Chapter Discussion** section examines the interconnections

among the Chapters, detailing how the frameworks were tested in **practical use cases**, particularly in energy forecasting, and demonstrating their effectiveness in **improving operational efficiency and decision-making**. Additionally, it explores the broader relevance of the frameworks across various sectors, highlighting the importance of **collaboration between AI specialists and domain experts** to enhance decision-making processes.

- The **Practical Takeaways** section distills the core findings into **actionable insights**, highlighting the significance of modular design, user-friendly interfaces, and the democratization of AI technologies.
- Finally, the **Research Perspectives** section outlines promising **future directions**, including the expansion of DSS frameworks into new application domains, advancements in real-time capabilities, and the integration of privacy and security measures.

By synthesizing these contributions, this chapter underscores the long-term significance of this research in shaping **the next generation of AI-driven Decision Support Systems**, fostering innovation, and enabling **more informed, transparent, and effective decision-making processes**.

5.1 Scientific Conclusions

This section consolidates the scientific contributions and key findings presented throughout this thesis. Each Chapter introduced a novel framework or methodology to address fundamental challenges in **DSS and AI pipeline design**. The core contributions of this work focused on enhancing data quality, improving AI accessibility, and fostering more effective decision-making processes. The advancements outlined in this thesis have been rigorously validated through real-world applications, demonstrating their practical impact and potential for broader adoption.

5.1.1 Advancing Data Quality in AI Systems

Chapter 2 presented an innovative **Data Quality Module (DQM) framework** designed for seamless integration into DSS. This module tackled the critical challenge of systematically evaluating input data quality and integrating preprocessing mechanisms into AI pipelines. The framework operates in three key stages:

- Aligning with **European data quality guidelines** through a rule-based mechanism.
- Assessing core **Data Quality (DQ) dimensions** such as completeness and consistency using well-defined mathematical formulations.
- Identifying and addressing **common Data Cleaning (DC) issues** through structured, pre-defined preprocessing instructions.

The module introduces **intuitive data quality indicators**, providing clear, quantitative insights into dataset reliability. These indicators serve as actionable metrics that empower analysts to make informed decisions about data preprocessing strategies, ensuring more robust and trustworthy AI models.

Validation was conducted through a proof-of-concept implementation using **energy time-series data from Belgium**, successfully demonstrating the module's effectiveness in detecting and rectifying data inconsistencies, missing values, and semantic anomalies. The **human-in-the-loop** approach further enriched the framework, incorporating user feedback to enhance adaptability and long-term accuracy.

The results underscore the transformative impact of the DQM framework on the streamlining of AI data preparation, significantly reducing preprocessing time while improving the clarity and usability of datasets. By integrating modularity and flexibility, this work represents a substantial leap forward in **improving data quality for AI-driven decision-making systems**.

5.1.2 Bridging AI Complexity and User Accessibility

Chapter 3 introduced **DSS4EX**, a novel DSS framework designed to **explore AI pipelines** and to make them more accessible and comprehensible to both technical and non-technical users. Unlike traditional DSS architectures, DSS4EX incorporates **interactive exploration features** that allow users to **engage with AI models dynamically** and make well-informed decisions.

The framework was successfully validated through a **real-world use case in electricity demand forecasting**, leveraging the **Decomposition-Residuals Deep Neural Network (DR-DNN) pipeline**. This prototype showcased how intuitive design choices, expert-guided recommendations, and interactive AI configurations can facilitate seamless AI adoption.

The findings of this study highlight **DSS4EX's ability to bridge the gap between AI complexity and practical usability**, making sophisticated

forecasting techniques accessible to a broader audience. Its modular and human-in-the-loop approach ensures transparency while enabling users with diverse levels of expertise to explore, adjust, and refine AI configurations effectively. This work contributes significantly to democratizing AI by promoting the wider adoption of decision-support tools tailored for real-world applications.

5.1.3 Expanding the Frontiers of AI-Powered Decision Support Systems

Chapter 4 introduced the **Explorative Decision Support System (EX-DSS)**, a groundbreaking extension of conventional DSS designed for the **design, deployment, and evaluation of AI pipelines**. This framework integrates two major enhancements:

- A **DQM**, ensuring high-quality data inputs for more reliable AI models.
- An **Explorative Management Subsystem**, providing project-specific recommendations and facilitating collaboration.

The **EX-DSS prototype** was applied to **smart plug load forecasting**, demonstrating its capability to optimize energy consumption in commercial buildings. Through an intuitive and modular interface, users could upload datasets, assess input quality, explore AI forecasting techniques, and deploy optimized pipelines. This implementation highlighted how EX-DSS **empowers users with both domain-specific insights and expert-driven guidance**, ensuring accurate forecasting and efficient decision-making.

Furthermore, EX-DSS introduced a **dual support system**, combining general knowledge through AI-driven guidance with project-specific recommendations from the Explorative Management Subsystem. The **human-in-the-loop** approach preserved user autonomy while integrating expert knowledge dynamically, enhancing both usability and analytical depth.

The results of this study confirm that **EX-DSS is an effective and versatile tool for AI pipeline design**, significantly reducing the complexity, time, and effort involved in deploying AI-driven forecasting solutions. Its emphasis on collaboration, data transparency, and reproducibility fosters a **sustainable research ecosystem** where users can share models, insights, and data-driven strategies, further amplifying its impact.

5.1.4 Final Reflections and Future Perspectives

The contributions presented in this thesis lay the groundwork for the next generation of **Decision Support Systems**, where AI pipeline configuration becomes more accessible, transparent, and efficient. The proposed frameworks address critical challenges in data quality, AI accessibility, and collaborative decision-making, making sophisticated AI solutions more practical and adaptable.

Despite these advancements, there are still opportunities for future exploration. Key areas for continued research include:

- Integrating **AI-driven automation** into DSS architectures while preserving human oversight.
- Expanding the frameworks to handle more diverse datasets and forecasting scenarios.
- Enhancing visualization capabilities within **EX-DSS** to allow for direct in-system result analysis and comparative pipeline evaluation.
- Strengthening collaborative functionalities to scale knowledge-sharing and reproducibility in AI-based decision-making.

Ultimately, the methodologies and innovations outlined in this thesis pave the way for more intelligent, adaptable, and user-centric **Decision Support Systems**. By enabling data-driven, high-quality AI pipelines, these contributions empower researchers, practitioners, and industries alike to make more informed, efficient, and impactful decisions. The future of DSS lies in bridging human expertise with **AI-driven intelligence**, and this work represents a significant step toward realizing that vision.

5.2 Cross-Chapter Discussions

5.2.1 General Conclusions

This thesis has successfully designed, implemented, and validated a series of frameworks that enhance **DSS for generating AI pipelines in energy time series applications**. By addressing critical challenges in data quality, AI accessibility, and decision support, this research has made substantial

contributions to both the theoretical foundations and practical implementations of **AI-driven DSS technologies**.

A key breakthrough of this work is the introduction of **DQM**, which represents a significant step forward in the preprocessing phase of AI pipeline development. By aligning with European Union guidelines and incorporating quantitative DQ metrics, this module provides users with structured, actionable insights to assess and improve dataset reliability. Ensuring that AI models are built on high-quality, well-prepared data leads to enhanced performance and greater predictive accuracy. Furthermore, this framework streamlines the data cleaning process, making it more efficient and accessible to users with varying levels of expertise.

Building upon this foundation, the **DSS4EX framework** extends DSS capabilities by offering an interactive, intuitive, and educational platform that simplifies the exploration and configuration of AI pipelines. Through expert-driven suggestions and guided interactions, DSS4EX bridges the gap between advanced AI methodologies and practical user applications, particularly in **time series forecasting**. Its design fosters greater transparency, usability, and trust, making AI-powered decision-making more approachable and reliable.

The culmination of this research is the **EX-DSS framework**, which integrates the **DQ module and DSS4EX** into a comprehensive and extensible DSS solution. EX-DSS extends traditional DSS architectures by supporting the entire lifecycle of AI pipeline design and deployment, from data quality assessment to model configuration, training, and evaluation. By embedding project-specific recommendations and a **human-in-the-loop approach**, EX-DSS ensures that users maintain complete control over AI-driven decision processes, fostering collaboration, adaptability, and user trust. Its modular and flexible architecture allows it to be adapted across diverse domains, further amplifying its potential impact.

The real-world applications explored in this thesis, such as **smart plug forecasting for energy optimization in commercial buildings**, highlight the practical relevance of these frameworks. These case studies demonstrate how the proposed solutions enable the efficient design and deployment of AI pipelines, optimizing resource use and contributing to global energy efficiency goals. Moreover, the findings confirm that these frameworks enhance transparency, reproducibility, and user engagement, which are key factors in the successful adoption of **AI-powered decision support systems**.

In a broader sense, this thesis makes a meaningful contribution to the democratization and trustworthiness of **AI-driven DSS**. By addressing key challenges related to data quality, user accessibility, and interaction, these

frameworks pave the way for broader adoption of AI technologies across various sectors, empowering organizations to make better, data-driven decisions.

Future research will focus on expanding the capabilities of these frameworks, including:

- **Real-time data processing** to support dynamic, adaptive decision-making.
- **Enhanced privacy-preserving mechanisms** to ensure secure AI adoption.
- **Application across additional domains**, such as healthcare, finance, and industrial automation.

By integrating these advancements, the proposed frameworks can continue to evolve into next-generation DSS solutions, enabling **more intelligent, transparent, and efficient AI-driven decision support systems**.

5.2.2 Real-World Applicability

The frameworks developed and presented in this thesis have demonstrated significant real-world applicability through their validation in **energy forecasting use cases**. These practical implementations underscore the robustness, adaptability, and operational benefits of the proposed systems, particularly in addressing challenges associated with time series data and AI pipeline configuration.

The **energy forecasting use cases** served as a critical testbed for the frameworks, showcasing their ability to optimize decision-making processes and enhance operational efficiency. For example, the application of the **EX-DSS framework** to **smart plug load forecasting** illustrated its capacity to streamline the design and deployment of AI pipelines. By providing tools for **data quality assessment**, project-specific recommendations, and interactive pipeline configuration, the system enabled users to design models that accurately predict energy consumption patterns, contributing to more efficient energy management in commercial buildings.

In particular, the frameworks facilitated the automation of complex tasks, such as preprocessing and model configuration, significantly reducing the time and effort required to set up AI pipelines. This automation was complemented by user-centric features, such as intuitive interfaces and expert-guided suggestions, which made the systems accessible to a wide range of users, including those without advanced technical expertise. This accessibility ensures that domain

experts and decision-makers can leverage AI technologies effectively, fostering innovation and informed decision-making.

Moreover, the results from these use cases validated the frameworks' ability to handle large-scale datasets and deliver actionable insights. The integration of modular components, such as the **Data Quality module** and the **Explorative Management Subsystem**, ensured that the system could adapt to the specific requirements of energy forecasting tasks. The outputs, ranging from data quality reports to optimized scheduling and predictive models, highlighted the frameworks' versatility and practical utility in improving operational outcomes.

The energy use cases also emphasized the importance of **trustworthiness and transparency** in AI-driven decision support. By incorporating **human-in-the-loop approaches** and enabling users to understand and control every stage of the pipeline configuration, the frameworks fostered trust in the system outputs. This is particularly relevant in **energy management**, where decisions based on AI predictions can have significant economic and environmental impacts.

In conclusion, the real-world applications of the frameworks validated their practical benefits in enhancing operational efficiency and decision-making processes. By addressing the specific needs of **energy forecasting** and demonstrating adaptability to complex use cases, the proposed frameworks provide a solid foundation for further expansion into **diverse domains** where **decision support systems** are crucial. These results underline the potential of the frameworks to **drive meaningful advancements** in AI-driven decision-making across various sectors.

5.2.3 Interdisciplinary Insights

The research presented in this thesis provides valuable interdisciplinary insights by emphasizing the importance of **modular and user-friendly designs** in DSS to democratize AI technologies. By bridging the gap between advanced AI methodologies and user accessibility, the frameworks contribute to making **AI-driven solutions** broadly applicable and impactful across various sectors.

The **modular design** of the proposed frameworks stands out as a key innovation, enabling flexibility and adaptability to diverse use cases. This modularity ensures that components, such as the **DQ module** and **EX-DSS**, can be tailored to the specific needs of different fields. For instance, while the focus of this work was energy forecasting, the underlying principles of these frameworks make them applicable to domains such as **healthcare, finance, logistics, and education**. This adaptability is crucial for addressing the unique challenges

posed by each sector, such as varying data types, decision-making requirements, and user expertise levels.

Equally important is the emphasis on **user-friendly designs**, which democratize access to AI technologies. The frameworks were intentionally developed with intuitive interfaces and guided workflows, reducing the technical barriers that often limit AI adoption among non-experts. This user-centered approach empowers decision-makers, educators, and other stakeholders to leverage AI capabilities without requiring in-depth technical knowledge. By simplifying interactions with complex AI systems, the frameworks promote broader adoption and foster innovation across industries.

Furthermore, the integration of **human-in-the-loop elements** enhances transparency and trust, ensuring that users can actively engage with the system and understand its outputs. This is particularly significant in interdisciplinary contexts, where collaboration between domain experts and technical teams is essential. The frameworks facilitate such collaboration by providing clear, actionable insights that bridge the knowledge gap between AI specialists and end-users.

The interdisciplinary applicability of these frameworks also reflects the growing importance of cross-sector collaboration in addressing complex societal challenges. Whether optimizing energy consumption, enhancing healthcare outcomes, or streamlining supply chain operations, the ability to apply **AI-driven DSS** effectively across disciplines underscores their **transformative potential**.

In summary, the **modular and user-friendly designs** of the frameworks presented in this thesis highlight the importance of interdisciplinary approaches in the development and application of DSS. By democratizing access to AI technologies and fostering collaboration across sectors, these insights contribute to the broader goal of leveraging AI for societal benefit, driving innovation, and addressing real-world challenges effectively.

5.3 Practical Takeaways

The practical takeaways from this research emphasize the **transformative potential** of the proposed frameworks in making advanced AI technologies more accessible, flexible, and impactful. By focusing on **modular design**, user-friendly interfaces, and interdisciplinary collaboration, these frameworks address key challenges in **decision support systems**, paving the way for broader adoption and innovative applications across diverse sectors. This section distills

the core insights into actionable points, highlighting their significance for both researchers and practitioners aiming to leverage **AI-driven solutions** in real-world contexts.

The main points of the takeaways are the following:

- **Modular Framework Design:**
  - Enables flexibility and customization for diverse applications across sectors such as **healthcare, finance, and logistics**.
  - Facilitates targeted adaptation to address domain-specific challenges and requirements.
- **User-Friendly Interfaces:**
  - Reduces technical barriers, making advanced AI technologies accessible to non-expert users.
  - Promotes broader adoption of AI tools by simplifying interactions and workflows.
- **Human-in-the-Loop Integration:**
  - Enhances transparency, user trust, and engagement with AI systems.
  - Encourages active participation from users, ensuring better alignment of outputs with real-world needs.
- **Cross-Sector Collaboration:**
  - Bridges the gap between AI specialists and domain experts, fostering effective interdisciplinary cooperation.
  - Supports shared understanding and actionable insights in team-based decision-making scenarios.
- **Democratization of AI Technologies:**
  - Enables stakeholders from various backgrounds to leverage AI capabilities without requiring extensive technical expertise.
  - Encourages innovation and broader application of **AI-driven DSS**.
- **Real-World Problem-Solving:**
  - Highlights the value of adaptable frameworks in addressing complex societal challenges, such as **energy optimization**, **healthcare improvements**, and operational efficiency.
  - Demonstrates the potential of **AI-driven DSS** in driving impactful, practical outcomes across disciplines.

5.4 Research Perspectives

This section outlines future research perspectives derived from this thesis, offering a **roadmap** for extending and refining its contributions. It highlights potential directions for expanding the applicability of the proposed frameworks, addressing emerging challenges, and advancing the integration of **AI-driven DSS** across diverse domains. By building upon the findings and methodologies established in this work, these research perspectives aim to foster innovation, adaptability, and broader impact in both academic and practical contexts.

5.4.1 Expanding Application Domains

Expanding the application domains of the proposed frameworks represents a critical avenue for future research to validate their adaptability and generalizability across various sectors. While these frameworks have demonstrated their effectiveness in **energy forecasting**, their **modularity** and **user-centric design** position them as valuable tools for addressing challenges in fields such as **healthcare** and **finance**. These sectors, characterized by distinct datasets and complex operational needs, could benefit significantly from **AI-driven DSS**.

In healthcare, the integration of DSS frameworks has the potential to transform clinical decision-making. Automating diagnostic processes, improving treatment planning, and streamlining data analysis are just a few applications where these frameworks could be impactful [54, 46]. One of the notable contributions of the **DQ module**, as outlined in Chapter 2, is its ability to assess and enhance the quality of input data. Applied to healthcare, this feature could address common issues such as missing or inconsistent information in electronic health records. Additionally, the ability to design AI pipelines tailored to specific healthcare applications could facilitate the development of predictive models for patient outcomes, improving care quality while adhering to stringent data security and privacy standards.

Similarly, the finance sector presents opportunities for applying DSS frameworks to improve risk assessment, fraud detection, and investment strategies. Financial datasets often require real-time processing and must deal with high variability and demand interpretable models [46]. The **EX-DSS** described in Chapter 4, with its focus on modularity and project-specific recommendations, could provide a robust foundation for creating AI pipelines in finance. Features such as automated data quality assessment and user-guided configuration could help address domain-specific challenges, enhancing predictive accuracy and enabling strategic insights that drive informed decision-making.

Expanding the frameworks into these diverse application domains not only validates their adaptability but also highlights the importance of user-friendly and modular designs, as emphasized in Chapter 3. These features ensure that non-expert users can interact effectively with the systems, broadening accessibility and promoting widespread adoption. Such democratization of AI technologies aligns with broader efforts to make AI tools impactful across various industries, ensuring they address societal needs beyond specialized technical fields.

5.4.2 Expanding Real-Time Capabilities

The future of **AI-driven DSS** lies in their ability to process and analyze information in real time, enabling rapid and intelligent decision-making in dynamic environments. Enhancing the proposed frameworks to support **real-time data analysis** is a crucial step toward this vision, particularly for applications in **autonomous systems**, **IoT-based monitoring**, and **high-frequency financial trading**. While the current designs have been successfully validated for batch data processing, their ability to handle streaming data remains an open challenge.

Real-time decision-making demands low-latency processing, continuous adaptation, and intelligent learning mechanisms. In IoT-driven environments, for example, sensors generate massive data streams that require on-the-fly analysis to detect anomalies, optimize resource allocation, or trigger automated responses. Similarly, autonomous vehicles depend on instantaneous AI-driven interpretations of environmental data to ensure safe and efficient navigation.

To address these demands, the proposed frameworks must evolve beyond static processing. As outlined in Chapter 4, reconfiguring the data management subsystems to support **high-velocity streaming data** is essential. This could be achieved through the integration of **online learning algorithms**, **incremental model updates**, and **event-driven architectures** that dynamically adapt to new information. Furthermore, enhancing system infrastructure with **distributed computing strategies**, such as edge computing for IoT applications or parallelized model inference, would ensure that real-time capabilities do not compromise performance or reliability.

By expanding into real-time analytics, these frameworks could transform AI-powered DSS into **autonomous, adaptive decision-making tools**, unlocking new possibilities in sectors where instantaneous insights are critical.

5.4.3 Integrating Privacy and Security

As **AI-driven DSS** become increasingly embedded in sensitive and high-stakes domains such as **healthcare, finance**, and smart infrastructure, ensuring **robust privacy and security mechanisms** is paramount. The frameworks proposed in this thesis must evolve to meet **stringent data protection standards** while maintaining their usability and accessibility.

Chapters 2 and 4 highlight the importance of **data confidentiality** and **regulatory compliance**, particularly under frameworks such as the **General Data Protection Regulation (GDPR)**. Future research should focus on integrating **privacy-preserving methodologies**, such as **federated learning**, **differential privacy**, and **homomorphic encryption**, to enable secure AI model training without exposing sensitive data. These techniques would allow institutions to collaborate on AI advancements while ensuring data remains protected and decentralized.

In addition to privacy, strengthening cybersecurity defenses within the DSS architecture is essential. Implementing **end-to-end encryption**, robust authentication protocols, and real-time anomaly detection would mitigate risks such as data breaches, adversarial attacks, and unauthorized access. The **modular design** of the proposed frameworks, as demonstrated throughout this thesis, provides a solid foundation for incorporating these security enhancements without compromising system usability.

By embedding **state-of-the-art security measures**, the frameworks can foster trust and confidence among users, ensuring their adoption in critical domains where **data integrity** and **confidentiality** are non-negotiable.

5.4.4 Automated Enhancements

The integration of **AutoML (Automated Machine Learning)** into the proposed frameworks represents a transformative opportunity to streamline AI pipeline configurations while maintaining user control and interpretability. Currently, **DSS4EX** (Chapter 3) emphasizes guided user interactions for pipeline design, but the next stage of evolution involves incorporating intelligent automation to further reduce time and expertise barriers.

AutoML has the potential to automate key AI development processes, including **model selection**, **hyperparameter tuning**, and **feature engineering**, making advanced AI solutions more accessible and efficient. However, full automation must be balanced with interpretability; users should understand, validate, and refine the AI-generated configurations. Future implementations

should explore **explainable AutoML techniques** that justify model decisions and provide users with clear rationales behind suggested configurations.

By integrating **adaptive AutoML pipelines**, these frameworks could significantly enhance scalability, usability, and efficiency, making **AI-driven DSS** a powerful tool for both domain experts and non-technical users across industries.

5.4.5 Educational and Training Tools

The democratization of AI requires more than just technical advancements; it necessitates empowering users with knowledge and hands-on experience. The current frameworks, particularly **DSS4EX** (Chapter 3), already prioritize user accessibility through intuitive interfaces and structured workflows. However, to further bridge the knowledge gap, future research should focus on integrating **interactive educational tools** directly within these systems.

A promising direction is the development of immersive learning modules, such as:

- **Step-by-step interactive tutorials** that guide users through AI pipeline creation and model evaluation.
- **Scenario-based simulations** where users can experiment with different data scenarios and decision strategies.
- **Real-time visualizations** of data flow, model performance, and decision-making pathways to enhance interpretability.

Furthermore, incorporating **domain-specific training programs**, such as AI applications in **healthcare diagnostics**, **financial forecasting**, or **industrial automation**, would allow users to engage with real-world case studies, strengthening their understanding of **AI-driven DSS applications**.

By embedding educational and training components, these frameworks could foster **AI literacy**, improve adoption rates, and bridge the gap between technical and non-technical stakeholders, ensuring that AI-driven decision support becomes a widely accessible and impactful technology.

5.4.6 Final Thoughts on Future Research

The research directions outlined above represent key areas where the proposed frameworks can be further refined, expanded, and applied. By advancing **real-**

time capabilities, **privacy mechanisms**, **automation**, and **educational tools**, these systems have the potential to become cornerstones in the evolution of **AI-powered decision support**. The future of DSS lies in creating systems that are intelligent yet interpretable, automated yet user-controlled, and powerful yet accessible.

The **modularity** and **flexibility** demonstrated throughout this thesis provide a strong foundation for these next steps. As AI continues to shape decision-making across industries, research must prioritize **scalability**, **security**, and **user engagement**, ensuring that AI-driven DSS are not only technically advanced but also **trusted**, **transparent**, and impactful in real-world applications.

Appendix A

DSS4EX: Block Descriptions

This appendix provides an illustration of the blocks utilized in the Decomposition Residual - Deep Neural Networks (DR-DNN) pipeline, as applied in the context described in Chapter 3.

Table A.1: **Block Descriptions in the DR-DNN Pipeline.** This table lists and describes each block used in the Decomposition Residual - Deep Neural Networks (DR-DNN) pipeline, which is presented in Chapter 3. The descriptions are shown to the user during the pipeline configuration phase to provide clarity on each block's function and purpose.

Block's Name	**Description**
Endogenous Signal	The endogenous signal represents the variable that the model predicts. In the examined case, it is the power load.
Exogenous Signals	Exogenous signals are independent variables that can help the model to make better predictions.
Robust Signal Decomposition (STL)	STL stands for Seasonal-Trend decomposition using LOWESS (locally weighted scatterplot smoothing). This method decomposes the raw signal into three distinct signals to squeeze out as much information as possible. Seasonality, Trend, and Residuals (remainder) are the three signals.

Continued on next page

Table A.1: **Block Descriptions in the DR-DNN Pipeline.** This table lists and describes each block used in the Decomposition Residual - Deep Neural Networks (DR-DNN) pipeline, which is presented in Chapter 3. The descriptions are shown to the user during the pipeline configuration phase to provide clarity on each block's function and purpose. (Continued)

Block's Name	**Description**
λ parameters from tests v	This block sets the p, d, q parameters found after majority voting of ν statistical tests.
Moving Filters	A moving filter is a technique that denoises the signals.
Lags	The signal is shifted backward for a certain amount of time. The variable p indicates the number of lags (hours, days, weeks).
Differencing	This block implements the differencing technique that transforms a non-stationary signal (a signal whose statistical properties, such as mean and variance, change over time) into a stationary one. It subtracts the current value of the signal from the previous one.
Time Signals	This block generates sinusoidal calendar features (sin/-cosines of day of week/month/year, weekend/weekday, day/night cycles, etc). These sinusoidal calendar features help the DNN model to capture consumption patterns in relation to time.
DNN Model	DNN stands for Deep Neural Network. A DNN is a neural network with more than two layers. Deep neural networks use sophisticated mathematical modeling to process data in complex ways. In this application, the only DNN model available is the Long Short-Term Memory network (LSTM), which is a type of recurrent neural network widely used in time series forecasting.
Time Series Analysis	This block contains traditional system identification tools that analyze signal correlations in time.
Feature Importance (SHAP)	SHAP stands for Shapley Additive explanations, and it is a game-theoretic approach that helps in understanding how the model behaves.

Continued on next page

Table A.1: **Block Descriptions in the DR-DNN Pipeline.** This table lists and describes each block used in the Decomposition Residual - Deep Neural Networks (DR-DNN) pipeline, which is presented in Chapter 3. The descriptions are shown to the user during the pipeline configuration phase to provide clarity on each block's function and purpose. (Continued)

Block's Name	**Description**
Forecast	This block presents, using plots, the predictions computed by the pipelines.

Appendix B

DSS4EX: Packages Requirements

This appendix outlines the software packages and their respective versions employed in developing the DSS4EX prototype, as presented in Chapter 3. These versions were carefully chosen to ensure compatibility, stability, and reproducibility, thereby upholding the integrity of the framework.

Table B.1: **Python Packages Used in the DSS4EX Framework.** This table lists the Python packages utilized in the development of the DSS4EX prototype, including their respective versions and descriptions. These packages were selected to ensure compatibility, stability, and reproducibility, which are essential for the framework's functionality, as detailed in Chapter 3. The table also includes links to the package documentation for reference.

Package Name	**Version**	**Description**
apache-airflow	2.7.3	Workflow automation and scheduling tool for managing complex workflows. `https://pypi.org/project/apache-airflow/`
dash	2.13.0	Web application framework for creating interactive, analytical applications. `https://pypi.org/project/dash/`

Continued on next page

Table B.1: **Python Packages Used in the DSS4EX Framework.** This table lists the Python packages utilized in the development of the DSS4EX prototype, including their respective versions and descriptions. These packages were selected to ensure compatibility, stability, and reproducibility, which are essential for the framework's functionality, as detailed in Chapter 3. The table also includes links to the package documentation for reference. (Continued)

Package Name	**Version**	**Description**
dash-bootstrap-components	1.4.1	Bootstrap components for styling Dash applications. `https://pypi.org/project/dash-bootstrap-components/`
dash-core-components	2.0.0	Core UI components for Dash applications, such as graphs and controls. `https://pypi.org/project/dash-core-components/`
dash-cytoscape	0.3.0	Graph visualization component for Dash applications. `https://pypi.org/project/dash-cytoscape/`
dash-daq	0.5.0	Data acquisition and control components for Dash. `https://pypi.org/project/dash-daq/`
dash-editor-components	0.0.2	Additional components for editing capabilities in Dash apps. `https://pypi.org/project/dash-editor-components/`
dash-html-components	2.0.0	HTML components for building Dash applications. `https://pypi.org/project/dash-html-components/`
Flask	2.2.2	Lightweight WSGI web application framework for Python. `https://pypi.org/project/Flask/`
matplotlib	3.8.2	Plotting library for creating static, animated, and interactive visualizations. `https://pypi.org/project/matplotlib/`
numpy	1.24.4	Fundamental package for numerical computing in Python. `https://pypi.org/project/numpy/`

Continued on next page

Table B.1: **Python Packages Used in the DSS4EX Framework.** This table lists the Python packages utilized in the development of the DSS4EX prototype, including their respective versions and descriptions. These packages were selected to ensure compatibility, stability, and reproducibility, which are essential for the framework's functionality, as detailed in Chapter 3. The table also includes links to the package documentation for reference. (Continued)

Package Name	**Version**	**Description**
openpyxl	3.1.2	Library for reading and writing Excel (.xlsx) files. `https://pypi.org/project/openpyxl/`
pandas	2.1.4	Data analysis and manipulation library. `https://pypi.org/project/pandas/`
scikit-learn	1.5.2	Machine learning library with simple and efficient tools. `https://pypi.org/project/scikit-learn/`
scikit-learn-extra	0.3.0	Additional machine learning algorithms for scikit-learn. `https://pypi.org/project/scikit-learn-extra/`
scipy	1.10.1	Library for scientific and technical computing. `https://pypi.org/project/scipy/`
seaborn	0.13.2	Data visualization library based on matplotlib. `https://pypi.org/project/seaborn/`
shap	0.44.0	Explainable AI library for interpreting model predictions. `https://pypi.org/project/shap/`
smm	0.1.5	Social Media Mining library, often used for text processing. `https://pypi.org/project/smm/`
statsmodels	0.14.0	Library for statistical models and hypothesis tests. `https://pypi.org/project/statsmodels/`
tensorflow	2.18.0	Deep learning framework developed by Google. `https://pypi.org/project/tensorflow/`

Continued on next page

Table B.1: **Python Packages Used in the DSS4EX Framework.** This table lists the Python packages utilized in the development of the DSS4EX prototype, including their respective versions and descriptions. These packages were selected to ensure compatibility, stability, and reproducibility, which are essential for the framework's functionality, as detailed in Chapter 3. The table also includes links to the package documentation for reference. (Continued)

Package Name	**Version**	**Description**
torch	2.1.2	Deep learning framework by Facebook, commonly used with PyTorch. `https://pypi.org/project/torch/`
torchx	0.7.0	Extensions and utilities for PyTorch development. `https://pypi.org/project/torchx/`

Appendix C

EX-DSS: Keywords

The keywords used to define datasets and projects in the application of the EX-DSS, as detailed in Chapter 4, are listed below.

- time series,
- energy,
- smart plugs,
- fast,
- high speed,
- less computation,
- accuracy,
- high performance,
- precise,
- reliable,
- user convenient,
- minimal disruption,
- user-friendly,
- non-intrusive.

Bibliography

[1] Ada, S., and Ghaffarzadeh, M. Decision making based on management information system and decision support system. *European Researcher 93* (03 2015), 260–269.

[2] Ahn, B., Kim, J., and Choi, B. Artificial intelligence-based machine learning considering flow and temperature of the pipeline for leak early detection using acoustic emission. *Engineering Fracture Mechanics 210* (2019), 381–392. Application of Acoustic Emission Techniques in Fracture Mechanics.

[3] Amershi, S., Arksey, N., Carenini, G., Conati, C., Mackworth, A., Maclaren, H., and Poole, D. Designing cispace: pedagogy and usability in a learning environment for ai. In *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education* (New York, NY, USA, 2005), ITiCSE '05, Association for Computing Machinery, p. 178–182.

[4] Asemi, A., Safari, A., and Asemi, A. The role of management information system (mis) and decision support system (dss) for manager's decision making process. *International Journal of Business and Management 6* (06 2011).

[5] Bammidi, T. R., Gutta, L. M., Kotagiri, A., Samayamantri, L. S., and Vaddy, R. K. The crucial role of data quality in automated decision-making systems. *International Journal of Management Education for Sustainable Development 7*, 7 (2024), 1–22.

[6] Beets, B., Newman, T., Howell, E., Bao, L., and Yang, S. Surveying public perceptions of artificial intelligence in health care in the united states: Systematic review. *Journal of medical Internet research 25* (04 2023), e40337.

[7] Blecker, S., Austrian, J. S., Horwitz, L. I., Kuperman, G., Shelley, D., Ferrauiola, M., and Katz, S. D. Interrupting providers with clinical decision support to improve care for heart failure. *International Journal of Medical Informatics 131* (2019).

[8] Botman, L., Lago, J., Fu, X., Chia, K., Wolf, J., Kleissl, J., and De Moor, B. Building plug load mode detection, forecasting and scheduling. *Applied Energy* (2024).

[9] Botman, L., Soenen, J., Theodorakos, K., Yurtman, A., Bekker, J., Vanthournout, K., Blockeel, H., Moor, B. D., and Lago, J. A scalable ensemble approach to forecast the electricity consumption of households. *IEEE Transactions on Smart Grid* (2022).

[10] Bragazzi, N., and Garbarino, S. Toward clinical generative ai: Conceptual framework. *JMIR AI* (06 2024).

[11] Brauner, P., Hick, A., Philipsen, R., and Ziefle, M. What does the public think about artificial intelligence?—a criticality map to understand bias in the public perception of ai. *Frontiers in Computer Science 5* (2023).

[12] Bukar, U., Sayeed, M. S., Abdul Razak, S. F., Yogarayan, S., and Sneesl, R. Decision-making framework for the utilization of generative artificial intelligence in education: A case study of chatgpt. *IEEE Access PP* (01 2024), 1–1.

[13] Campitelli, G., and Gobet, F. Herbert simon's decision-making approach: Investigation of cognitive processes in experts. *Review of General Psychology 14* (12 2010), 354–364.

[14] Carrasco Ramírez, J. G., and Islam, M. Application of artificial intelligence in practical scenarios. *Journal of Artificial Intelligence General Science (JAIGS)* (2024).

[15] Cave, S., Coughlan, K., and Dihal, K. "scary robots": Examining public responses to ai. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society* (New York, NY, USA, 2019), AIES '19, Association for Computing Machinery, p. 331–337.

[16] Chandra, A., Krovi, R., and Rajagopalan, B. Risk visualization: A mechanism for supporting unstructured decision making processes. *The International Journal of Applied Management and Technology* (2007).

[17] Che, C., Huang, Z., Li, C., Zheng, H., and Tian, X. Integrating generative ai into financial market prediction for improved decision making. *Applied and Computational Engineering 64* (05 2024), 156–162.

[18] Chengalur-Smith, I., Ballou, D., and Pazer, H. The impact of data quality information on decision making: An exploratory analysis. *Knowledge and Data Engineering, IEEE Transactions on 11* (12 1999), 853 – 864.

[19] Chengalur-Smith, I., Ballou, D., and Pazer, H. The impact of data quality information on decision making: an exploratory analysis. *IEEE Transactions on Knowledge and Data Engineering* (1999).

[20] Chia, K., Lebar, A., Agarwal, V., Lee, M., Ikedo, J., Wolf, J., Trenbath, K., and Kleissl, J. Integration of a Smart Outlet-Based Plug Load Management System with a Building Automation System. *2023 IEEE PES Grid Edge Technologies Conference and Exposition, Grid Edge 2023* (2023).

[21] Clement, T., Kemmerzell, N., Abdelaal, M., and Amberg, M. Xair: A systematic metareview of explainable ai (xai) aligned to the software development process. *Machine Learning and Knowledge Extraction 5*, 1 (2023), 78–108.

[22] Data Europa EU. *Data Quality Guidelines*. Publications Office of the European Union, 2021.

[23] De Angelis, L., Baglivo, F., Arzilli, G., Privitera, G. P., Ferragina, P., Tozzi, A. E., and Rizzo, C. Chatgpt and the rise of large language models: the new ai-driven infodemic threat in public health. *Frontiers in public health* (2023).

[24] De Silva, D., and Alahakoon, D. An artificial intelligence life cycle: From conception to production. *Patterns 3*, 6 (2022), 100489.

[25] Dellermann, D., Lipusch, N., Ebel, P., and Leimeister, J. M. Design principles for a hybrid intelligence decision support system for business model validation. *Electronic Markets 29* (2019), 423–441.

[26] Dol, S. M., and Jawandhiya, P. M. Use of data mining tools in educational data mining. In *2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT)* (2022), pp. 380–387.

[27] Duan, Y., and Xu, M. Decision support systems in small businesses. *IGI Global - Business and Information Systems Research Centre (BISC)* (2009).

[28] Ehrlinger, L., Haunschmid, V., Palazzini, D., and Lettner, C. A daql to monitor data quality in machine learning applications. In

Prooceedings of the 30th International Conference on Database and Expert Systems Applications - Part I (2019).

[29] Ehrlinger, L., and Wöss, W. Automated data quality monitoring. In *Proceedings of the 22nd MIT International Conference on Information Quality (ICIQ 2017)* (2017).

[30] Ehrlinger, L., and Wöss, W. A survey of data quality measurement and monitoring tools. *Frontiers in Big Data* (2022).

[31] Elhaddad, M., and Hamam, S. Ai-driven clinical decision support systems: An ongoing pursuit of potential. *Cureus 16* (04 2024).

[32] Elragal, A., and Elgendy, N. Design principles supporting data-driven decisions platforms. *Research Square* (2024).

[33] Enholm, I. M., Papagiannidis, E., Mikalef, P., and Krogstie, J. Artificial Intelligence and Business Value: a Literature Review. *Information Systems Frontiers 24*, 5 (October 2022), 1709–1734.

[34] Eom, S. Reference disciplines of decision support systems. *Handbook on Decision Support Systems 1 Basic Themes* (01 2008).

[35] Eom, S., and Kim, E. A survey of decision support system applications (1995–2001). *Journal of the Operational Research Society 57* (11 2006), 1264–1278.

[36] Er, M. Decision support systems: A summary, problems, and future trends. *Decision Support Systems 4*, 3 (1988), 355–363.

[37] Eurostat. Energy statistics - supply, transformation and consumption. `https://www.eea.europa.eu/data-and-maps/data/external/supply-transformation-consumption-electricity-annual-data`, 2020.

[38] Farrokhabadi, M., Browell, J., Wang, Y., Makonin, S., Su, W., and Zareipour, H. Day-ahead electricity demand forecasting competition: Post-covid paradigm. *IEEE Open Access Journal of Power and Energy 9* (2022), 185–191.

[39] Fernando, J., and Baldelovar, M. Decision support system: Overview, different types and elements. *TechnoareteTransactions on Intelligent Data Mining and Knowledge Discovery 2* (05 2022).

[40] Feurer, M., Eggensperger, K., Falkner, S., Lindauer, M., and Hutter, F. Auto-sklearn 2.0: Hands-free automl via meta-learning. *Journal of Machine Learning Research 23*, 261 (2022), 1–61.

[41] Fisher, C., Chengalur-Smith, I., and Ballou, D. The impact of experience and time on the use of data quality information in decision making. *Information Systems Research 14* (06 2003), 170–188.

[42] Ghasemaghaei, M., and Calic, G. Can big data improve firm decision quality? the role of data quality and data diagnosticity. *Decision Support Systems 120* (2019), 38–49.

[43] Gulser, G., and Badur, B. Developing a framework for integrating knowledge management and decision support systems: Application to time series forecasting. *Communications of The IbIMA* (2011), 1–15.

[44] Haug, A., Zachariassen, F., and van Liempd, D. The costs of poor data quality. *Journal of Industrial Engineering and Management (JIEM) 4*, 2 (2011), 168–193.

[45] Heidrich, B., Bartschat, A., Turowski, M., Neumann, O., Phipps, K., Meisenbacher, S., Schmieder, K., Ludwig, N., Mikut, R., and Hagenmeyer, V. pywatts: Python workflow automation tool for time series, 2021.

[46] Holzinger, A. *The Next Frontier: AI We Can Really Trust.* Springer, 01 2021, pp. 427–440.

[47] Ignatiev, A. Towards trustable explainable ai. In *IJCAI* (2020), pp. 5154–5158.

[48] Jacquet-Lagreze, E., and Shakun, M. F. Decision support systems for semi-structured buying decisions. *European Journal of Operational Research* (1984).

[49] Jarrett, D., Yoon, J., Bica, I., Qian, Z., Ercole, A., and van der Schaar, M. Clairvoyance: A pipeline toolkit for medical time series. In *International Conference on Learning Representations* (2021).

[50] Kiefer, C. Quality indicators for text data. In *BTW 2019 – Workshopband* (2019).

[51] Kieseberg, P., Weippl, E., Tjoa, A. M., Cabitza, F., Campagner, A., and Holzinger, A. *Controllable AI - An Alternative to Trustworthiness in Complex AI Systems?* Springer Nature Switzerland, 08 2023, pp. 1–12.

[52] Kim, E., and Eom, S. Decision support systems application development trends (2002 – 2012). *International Journal of Information Systems in the Service Sector 8* (04 2016), 1–13.

[53] Li, B., Qi, P., Liu, B., Di, S., Liu, J., Pei, J., Yi, J., and Zhou, B. Trustworthy ai: From principles to practices. *ACM Comput. Surv. 55*, 9 (Jan. 2023).

[54] Liang, W., Tadesse, G. A., Ho, D., Li, F.-F., Zaharia, M., Zhang, C., and Zou, J. Advances, challenges and opportunities in creating data for trustworthy ai. *Nature Machine Intelligence 4* (08 2022).

[55] Lim, B., and Zohren, S. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 379*, 2194 (2021).

[56] Lindauer, M., Karl, F., Klier, A., Moosbauer, J., Tornede, A., Mueller, A. C., Hutter, F., Feurer, M., and Bischl, B. Position: A call to action for a human-centered AutoML paradigm. In *Proceedings of the 41st International Conference on Machine Learning* (21–27 Jul 2024), R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, Eds., vol. 235 of *Proceedings of Machine Learning Research*, PMLR, pp. 30566–30584.

[57] Liu, X., Ni, Z., Yuan, D., Jiang, Y., Wu, Z., Chen, J., and Yang, Y. A novel statistical time-series pattern based interval forecasting strategy for activity durations in workflow systems. *Journal of Systems and Software 84*, 3 (2011), 354–376.

[58] Longo, L., Brcic, M., Cabitza, F., Choi, J., Confalonieri, R., Ser, J. D., Guidotti, R., Hayashi, Y., Herrera, F., Holzinger, A., Jiang, R., Khosravi, H., Lecue, F., Malgieri, G., Páez, A., Samek, W., Schneider, J., Speith, T., and Stumpf, S. Explainable artificial intelligence (xai) 2.0: A manifesto of open challenges and interdisciplinary research directions. *Information Fusion 106* (2024), 102301.

[59] Maarif, M. R., Saleh, A. R., Habibi, M., Fitriyani, N. L., and Syafrudin, M. Energy usage forecasting model based on long short-term memory (lstm) and explainable artificial intelligence (xai). *Information 14*, 5 (2023).

[60] Mahajan, G., and Saini, B. Educational data mining: A state-of-the-art survey on tools and techniques used in edm. *International Journal of Computer Applications & Information Technology 12* (04 2020).

[61] Mangat, P. K., and Saini, D. K. S. Chapter 26 - relevance of data mining techniques in real life. In *System Assurances*, P. Johri, A. Anand, J. Vain, J. Singh, and M. Quasim, Eds., Emerging Methodologies and Applications in Modelling. Academic Press, 2022, pp. 477–502.

[62] Meisenbacher, S., Turowski, M., Phipps, K., Rätz, M., Müller, D., Hagenmeyer, V., and Mikut, R. Review of automated time series forecasting pipelines. *WIREs Data Mining and Knowledge Discovery 12*, 6 (Aug. 2022).

[63] Mintzberg, H. he new science of management decision, revised edition. *Administrative Science Quarterly 22*, 2 (1977), 342–351.

[64] Morison, J., and Mcinerney, T. When should a computer decide? judicial decision-making in the age of automation, algorithms and generative artificial intelligence. *Research Handbook on Judging and the Judiciary (Elgar-Routledge, Law and Society Series) S. Turenne and M. Moussa Eds., Available at SSRN* (03 2024).

[65] Nagahisarchoghaei, M., Nur, N., Cummins, L., Nur, N., Karimi, M. M., Nandanwar, S., Bhattacharyya, S., and Rahimi, S. An empirical survey on explainable ai technologies: Recent trends, use-cases, and categories from technical and application perspectives. *Electronics 12*, 5 (2023).

[66] Ojedokun, S., Samson, A., Olukoyejo, O., and Alatishe, T. Perceptions and opinions of medical professionals on artificial intelligence in optimizing the healthcare sector. *Asian Journal of Medical Principles and Clinical Practice 1* (05 2024), 279–288.

[67] Oliveira, O., and Oliveira, B. An extensible framework for data reliability assessment. In *Proceedings of the 24th International Conference on Enterprise Information Systems - Volume 1: ICEIS,* (2022).

[68] Ozdemir, D., Ayvaci, M., and Zentner, A. What people think of machines as doctors: Unveiling the value of gen-ai for e-health. *SRN Electronic Journal* (2024).

[69] Patel, K., Ramamurthy, P., Garg, P., Kumar, S., and Beeram, D. Ai-enhanced design: Revolutionizing methodologies and workflows. *INTERNATIONAL JOURNAL OF ARTIFICIAL INTELLIGENCE RESEARCH AND DEVELOPMENT (IJAIRD) 2*, 1 (06 2024), 135–157.

[70] Pelekis, S., Pountridis, T., Kormpakis, G., Lampropoulos, G., Karakolis, E., Mouzakitis, S., and Askounis, D. Deeptsf: Codeless machine learning operations for time series forecasting. *SoftwareX 27* (2024), 101758.

[71] Phillips-Wren, G. *Intelligent Decision Support Systems.* John Wiley & Sons, Ltd, 2013, ch. 2, pp. 25–44.

[72] Pigni, F., and Piccoli, G. *Information Systems for Managers, 4th Edition*. Prospect Press, 01 2019.

[73] Power, D. J. *Decision Support Systems: A Historical Overview*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 121–140.

[74] Power, D. M. J. A brief history of decision support systems. In *The Web Conference* (2003).

[75] Rane, J., Mallick, S., Kaya, Ö., and Rane, N. Automated machine learning (automl) in industry 4.0, 5.0, and society 5.0: Applications, opportunities, challenges, and future directions. *DeepScience Publishing* (10 2024), 181–206.

[76] Rashidi, M., Ghodrat, M., Samali, B., and Mohammadi, M. Decision support systems. In *Management of Information Systems*, M. Pomffyova, Ed. IntechOpen, Rijeka, 2018, ch. 2.

[77] Rezig, E. K., Ouzzani, M., Elmagarmid, A. K., Aref, W. G., and Stonebraker, M. Towards an end-to-end human-centric data cleaning framework. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics* (New York, NY, USA, 2019), HILDA '19, Association for Computing Machinery.

[78] Rinaldi, G., Crema Garcia, F., Agudelo, O., Becker, T., Vanthournout, K., Mestdagh, W., and De Moor, B. A framework for a data quality module in decision support systems: An application with smart grid time series. In *Proceedings of the 25th International Conference on Enterprise Information Systems - Volume 1: ICEIS* (2023), INSTICC, SciTePress, pp. 443–452.

[79] Rinaldi, G., Theodorakos, K., Crema Garcia, F., Agudelo, O. M., and De Moor, B. Dss4ex: A decision support system framework to explore artificial intelligence pipelines with an application in time series forecasting. *Expert Systems with Applications 269* (2025), 126421.

[80] Romero, C., and Ventura, S. Educational data mining and learning analytics: An updated survey. *WIREs Data Mining and Knowledge Discovery 10*, 3 (Jan. 2020).

[81] Rupnik, R., Kukar, M., Vračar, P., Košir, D., Pevec, D., and Bosnić, Z. Agrodss: A decision support system for agriculture and farming. *Computers and Electronics in Agriculture* (2019).

[82] Sadiq, S., and Indulska, M. Open data: Quality over quantity. *International Journal of Information Management* (2017).

[83] Saeed, M. Security, privacy, and robustness for trustworthy ai systems: A review. *Computers & Electrical Engineering 119* (09 2024).

[84] Salehin, I., Islam, M. S., Saha, P., Noman, S., Tuni, A., Hasan, M. M., and Baten, M. A. Automl: A systematic review on automated machine learning with neural architecture search. *Journal of Information and Intelligence 2*, 1 (2024), 52–81.

[85] Schwabe, D., Becker, K., Seyferth, M., Klass, A., and Schaeffter, T. The metric-framework for assessing data quality for trustworthy ai in medicine: a systematic review. *npj Digital Medicine 7* (08 2024).

[86] Sharda, R., Delen, D., and Turban, E. *Business Intelligence and Analytics: Systems for Decision Support*. Pearson Education, 2014.

[87] Shchur, O., Turkmen, C., Erickson, N., Shen, H., Shirkov, A., Hu, T., and Wang, Y. Autogluon-timeseries: Automl for probabilistic time series forecasting, 2023.

[88] Shim, J., Warkentin, M., Courtney, J., Power, D., Sharda, R., and Carlsson, C. Past, present, and future of decision support technology. *Decision Support Systems 33* (06 2002), 111–126.

[89] Sidi, F., Hassany, P., Affendey, L., A. Jabar, M., Ibrahim, H., and Mustapha, A. Data quality: A survey of data quality dimensions. In *2012 International Conference on Information Retrieval & Knowledge Management* (08 2012), pp. 300–304.

[90] Sjöström, J., Dryselius, P., Nygren, J., Nair, M., Soliman, A., and Lundgren, L. E. Design principles for machine learning based clinical decision support systems: A design science study. In *Design Science Research for a Resilient Future (DESRIST 2024)* (2024), vol. 14621 of *Lecture Notes in Computer Science*, Springer, pp. 109–122.

[91] Slater, S., Joksimovic, S., Kovanovic, V., Baker, R., and Gasevic, D. Tools for educational data mining: A review. *Journal of Educational and Behavioral Statistics 42* (01 2017), 85–106.

[92] Sloane, E. B., and J. Silva, R. Artificial intelligence in medical devices and clinical decision support systems. In *Clinical Engineering Handbook (Second Edition)*, E. Iadanza, Ed., second edition ed. Academic Press, 2020.

[93] Soenen, J., Yurtman, A., Becker, T., D'hulst, R., Vanthournout, K., Meert, W., and Blockeel, H. Scenario generation

of residential electricity consumption through sampling of historical data. *Sustainable Energy, Grids and Networks* (2023).

[94] Soori, M., Jough, F. K. G., Dastres, R., and Arezoo, B. Ai-based decision support systems in industry 4.0, a review. *Journal of Economy and Technology* (2024).

[95] Sprague, R. H. A framework for the development of decision support systems. *MIS Quarterly 4*, 4 (1980), 1–26.

[96] Stamenković, S., Jovanović, N., Vasović, B., Cvjetković, M., and Jovanović, Z. Software tools for learning artificial intelligence algorithms. *Artif. Intell. Rev. 56*, 9 (feb 2023), 10297–10326.

[97] Steidl, M., Felderer, M., and Ramler, R. The pipeline for the continuous development of artificial intelligence models—current state of research and practice. *Journal of Systems and Software 199* (2023), 111615.

[98] Storey, V. C., Hevner, A. R., and Yoon, V. Y. The design of human-artificial intelligence systems in decision sciences: A look back and directions forward. *Decision Support Systems 182* (2024), 114230.

[99] Sutton, R., Pincock, D., Baumgart, D., Sadowski, D., Fedorak, R., and Kroeker, K. An overview of clinical decision support systems: benefits, risks, and strategies for success. *NPJ Digit Med 3* (02 2020).

[100] Taye, M. M. Understanding of machine learning with deep learning: Architectures, workflow, applications and future directions. *Computers 12*, 5 (2023).

[101] Territo, K., and Romagnoli, J. Fastman-jmp: All-in-one tool for data mining and model building. In *34th European Symposium on Computer Aided Process Engineering / 15th International Symposium on Process Systems Engineering*, F. Manenti and G. V. Reklaitis, Eds., vol. 53 of *Computer Aided Chemical Engineering*. Elsevier, 2024, pp. 3421–3426.

[102] Theodorakos, K., Agudelo, O. M., Espinoza, M., and De Moor, B. Decomposition-residuals neural networks: Hybrid system identification applied to electricity demand forecasting. *IEEE Open Access Journal of Power and Energy 9* (2022), 241–253.

[103] Toreini, E., Aitken, M., Coopamootoo, K. P. L., Elliott, K., Zelaya, C. G., and van Moorsel, A. The relationship between trust in AI and trustworthy machine learning technologies. *CoRR abs/1912.00782* (2019).

[104] Turban, E., Aronson, J. E., and Liang, T.-P. *Decision Support Systems and Intelligent Systems (7th Edition)*. Prentice-Hall, Inc., USA, 2004.

[105] Turban, E., Sharda, R., and Delen, D. *Decision Support and Business Intelligence Systems*, 9th ed. Prentice Hall Press, 2010.

[106] Tuttle, R., Trenbath, K., Maisha, K., and LeBar, A. Assessing and reducing plug and process loads in office buildings. *Technical Report* (2020).

[107] Wang, Y.-C., Chen, T.-C., and Chiu, M.-C. An improved explainable artificial intelligence tool in healthcare for hospital recommendation. *Healthcare Analytics 3* (02 2023), 100147.

[108] Westergaard, G., Erden, U., Mateo, O. A., Lampo, S. M., Akinci, T. C., and Topsakal, O. Time series forecasting utilizing automated machine learning (automl): A comparative analysis study on diverse datasets. *Information 15*, 1 (2024).

[109] Whang, S. E., Roh, Y., Song, H., and Lee, J.-G. Data collection and quality challenges in deep learning: a data-centric ai perspective. *The VLDB Journal 32*, 4 (Jan. 2023), 791–813.

[110] Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J., Santos, L. O. B. d. S., Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T. W., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., González-Beltrán, A., Gray, A. J. G., Groth, P., Goble, C., Grethe, J. S., Heringa, J., Hoen, P. A., Hooft, R., Kuhn, T., Kok, R., Kok, J. N., Lusher, S. J., Martone, M. E., Mons, A., Packer, A. L., Persson, B. L., Rocca-Serra, P., Roos, M., Schaik, R. v., Sansone, S., Schultes, E., Sengstag, T., Slater, T., Strawn, G., Swertz, M. A., Thompson, M., Lei, J. v. d., Mulligen, E. M. v., Velterop, J., Waagmeester, A., Wittenburg, P., Wolstencroft, K., Zhao, J., and Mons, B. The fair guiding principles for scientific data management and stewardship. *Scientific Data* (2016).

[111] Windmann, A., Wittenberg, P., Schieseck, M., and Niggemann, O. Artificial intelligence in industry 4.0: A review of integration challenges for industrial systems, 2024.

[112] Wu, J., Wang, H., Ni, C., Zhang, C., and Lu, W. Data pipeline training: Integrating automl to optimize the data flow of machine learning

models. In *2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE)* (2024), pp. 730–734.

[113] Zhang, G., Lu, J., and Gao, Y. *Multi-Level Decision Making: Models, Methods and Applications*. Springer Publishing Company, Incorporated, 2015.

[114] Zhang, Y., Sun, Q., Qi, D., Liu, J., Ma, R., and Petrosian, O. *ShapTime: A General XAI Approach for Explainable Time Series Forecasting*. Springer, 01 2024, pp. 659–673.

[115] Zhou, C., Kuang, D., Liu, J., Yang, H., Zhang, Z., Mackworth, A., and Poole, D. Aispace2: An interactive visualization tool for learning and teaching artificial intelligence. *Proceedings of the AAAI Conference on Artificial Intelligence 34*, 09 (Apr. 2020), 13436–13443.

[116] Zielinska, K. Designing ai clinical decision support systems with a human-centric usability focus: Designs ai-driven clinical decision support systems with a focus on user-centered design principles to enhance usability and adoption. *Journal of AI-Assisted Scientific Discovery* (2024).

List of publications

International Conference Papers

Rinaldi, Giulia; Crema Garcia, Fernando; Agudelo, Oscar Mauricio; Becker, Thijs; Vanthournout, Koen; De Moor, Bart
"A Framework for a Data Quality Module in Decision Support Systems: an application with Smart Grid Time series"
ICEIS 2023 - The 25th International Conference on Enterprise Information Systems, Prague (Czech Republic), April 2023
DOI: 10.5220/0011749700003467
https://www.scitepress.org/Link.aspx?doi=10.5220/0011749700003467

Rinaldi, Giulia; Botman, Lola; Agudelo, Oscar Mauricio; De Moor, Bart
"EX-DSS: An Explorative Decision Support System for designing and deploying Smart Plug Forecasting Pipelines"
KMIS 2024 - The 16th International Conference on Knowledge Management and Information Systems, Porto (Portugal), November 2024
DOI: 10.5220/0012925800003838.
https://www.scitepress.org/Papers/2024/129258/129258.pdf

International Journal Papers

Rinaldi, Giulia; Theodorakos Konstantinos; Crema Garcia, Fernando; Agudelo, Oscar Mauricio; De Moor, Bart
"DSS4EX: A Decision Support System Framework to Explore Artificial Intelligence Pipelines with an application in Time Series Forecasting".
Expert Systems with Applications, ISSN: 0957-4174, 2025
DOI: 10.1016/j.eswa.2025.126421
https://www.sciencedirect.com/science/article/pii/S0957417425000430

Andhari, Madhavi Dipak; Rinaldi, Giulia; Nazari, Pouya; Vets, Johanna; Shankar, Gautam; Dubroja, Nikolina; Ostyn, Tessa; Vanmechelen, Maxime;

Decraene, Brecht; Arnould, Alexandre; Mestdagh, Willem; De Moor, Bart; De Smet, Frederik; Bosisio, Francesca; Antoranz, Asier
"Quality control of immunofluorescence images using artificial intelligence' .
Cell Reports Physical Science, ISSN: 2666-3864, 2024
DOI: 10.1016/j.xcrp.2024.102220
https://www.sciencedirect.com/science/article/pii/S2666386424005137

Poster presentations at international Conferences

Rinaldi, Giulia; Crema Garcia, Fernando; Agudelo, Oscar Mauricio; Becker, Thijs; Vanthournout, Koen; De Moor, Bart
"A Framework for a Data Quality Module in Decision Support Systems: an application with Smart Grid Time series"
ICEIS 2023 - The 25th International Conference on Enterprise Information Systems, Prague (Czech Republic), April 2023

Rinaldi, Giulia; Botman, Lola; Agudelo, Oscar Mauricio; De Moor, Bart
"EX-DSS: An Explorative Decision Support System for designing and deploying Smart Plug Forecasting Pipelines"
KMIS 2024 - The 16th International Conference on Knowledge Management and Information Systems, Porto (Portugal), November 2024
Awarded Best Poster

International Talks

Rinaldi, Giulia; Botman, Lola; Agudelo, Oscar Mauricio; De Moor, Bart
"A Decision Support System for Evaluating Smart Plug Forecasting Pipelines"
44th International Symposium on Forecasting, Dijon (France) July 202

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING
STADIUS
Kasteelpark Arenberg 10, 3001 Leuven
B-3001 Leuven
first.last@kuleuven.be
https://www.esat.kuleuven.be/stadius