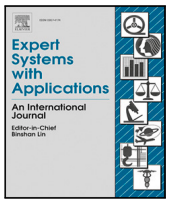
Expert Systems With Applications 269 (2025) 126421



Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



DSS4EX: A Decision Support System framework to explore Artificial Intelligence pipelines with an application in time series forecasting

Giulia Rinaldi *, Konstantinos Theodorakos, Fernando Crema Garcia, Oscar Mauricio Agudelo, Bart De Moor [1]

KU Leuven, Department of Electrical Engineering (ESAT), STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics, Kasteelpark Arenberg 10, Leuven, 3001, Belgium

ARTICLE INFO

Keywords:
Decision Support System
Pipeline exploration
Framework
Software architecture
Neural networks
Time series forecasting

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.

1. Introduction

Even though the term Decision Support System (DSS) has been used since the 1980s (Sprague, 1980), 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 (Blecker et al., 2019) and supports personalized treatment, prediction, and information gathering (Jarrett et al., 2021). In the industrial sector, DSS frameworks help convert knowledge into new models, often by building on existing ones (Gulser & Badur, 2011). For example, data mining tools have been integrated into DSS for farm management (Rupnik et al., 2019). 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 (Nagahisarchoghaei et al., 2023). 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 paper 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:

* Correspondence to: Kasteelpark Arenberg 10, Leuven, Belgium.
E-mail addresses: giulia.rinaldi@kuleuven.be (G. Rinaldi), konstantinos.theodorakos@kuleuven.be (K. Theodorakos), fernando.cremagarcia@kuleuven.be (F. Crema Garcia), mauricio.agudelo@kuleuven.be (O.M. Agudelo), bart.demoor@kuleuven.be (B. De Moor).
[1] Fellow IEEE, SIAM, IFAC

https://doi.org/10.1016/j.eswa.2025.126421
Received 20 August 2024; Received in revised form 22 December 2024; Accepted 3 January 2025
Available online 14 January 2025
0957-4174/© 2025 Published by Elsevier Ltd.

- **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 paper 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,[2] Knime,[3] and IBM SPSS,[4] 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. 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 paper is organized as follows. Section 2 discusses state-of-the-art tools for AI pipeline training, AI concepts, and DSS frameworks. Section 3 outlines the research objectives and challenges. Section 4 presents the DSS4EX framework. Section 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 (Theodorakos et al., 2022). Section 6 discusses the results and findings. Finally, Section 7 provides concluding remarks and future research directions.

2. Literature review

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 (Salehin et al., 2024). Tools like PyCaret help automate data preprocessing, model selection, and hyperparameter tuning, enabling non-experts to use complex techniques without extensive technical knowledge (Wu et al., 2024). Studies on AutoML tools like AutoGluon (Shchur et al., 2023), Auto-Sklearn (Feurer et al., 2022), and PyCaret[5] have shown varied results across different datasets, underscoring the importance of dataset-specific considerations (Westergaard et al., 2024). 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 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),[6] to enhance model transparency (Pelekis et al., 2024). 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 (Territo & Romagnoli, 2024). 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

[2] https://altair.com/altair-rapidminer.
[3] https://www.knime.com/.
[4] https://www.ibm.com/spss.
[5] https://pycaret.org/.
[6] https://shap.readthedocs.io/.

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.

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 (Carrasco Ramírez & Islam, 2024). As AI becomes more common, tools that help the general public understand 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 (De Angelis et al., 2023). 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 (Brauner et al., 2023; Cave et al., 2019). Even experts, such as medical doctors, use AI with caution due to its opacity (Beets et al., 2023; Ojedokun et al., 2024; Ozdemir et al., 2024). Understanding AI can facilitate the adoption of these AI-augmented systems (Longo et al., 2024). Educational software tools play a critical role, although they often target academic settings and are not problem-driven (Stamenković et al., 2023). Tools like AISpace2 and CIspace help in understanding general AI concepts but are limited in their application scope (Amershi et al., 2005; Zhou et al., 2020). 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 (Ignatiev, 2020). XAI is essential for creating trustworthy AI systems. In time series forecasting, tools like those described by Maarif et al. (2023) combine LSTM neural networks with XAI techniques to forecast energy consumption and provide interpretability using SHapley Additive exPlanations (SHAP). Zhang et al. (2024) 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) described an XAI system for hospital recommendations that enhances transparency through various visualization techniques. Clement et al. (2023) 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 paper 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.

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 (Er, 1988; Joey & Baldelovar, 2022).

A robust DSS framework includes several key components: a Data base 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 (Turban et al., 2010). 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) 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) 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 and Elgendy (2024) 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 (2024) 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.

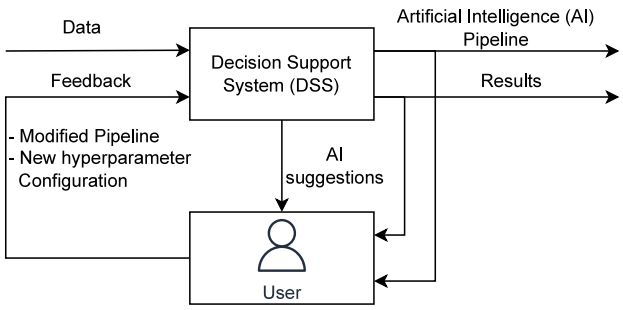


Fig. 1. High-Level Overview of the proposed framework. The input to the Decision Support System (DSS) is the data. The outputs are the Artificial Intelligence (AI) pipeline customized by the user, and the results are obtained by running it with the provided data. These outputs, along with AI suggestions, 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.

3. Research objectives/challenges

Recent advancements in software and technology aim to simplify computations and assist experienced researchers in data science. However, making AI and complex systems accessible to a broader audience, including those with less experience, remains challenging. We propose DSS4EX, a Decision Support System (DSS) framework designed to facilitate the exploration and understanding of AI pipelines for both experts and non-experts.

The framework addresses two main challenges: first, the complexity of AI pipelines that can be difficult for non-experts to grasp; second, the need for effective communication and information exchange between system components. To overcome these challenges, DSS4EX outlines clear communication protocols and information exchanges among internal components.

To demonstrate the benefits of DSS4EX, we developed a software demo tailored to a real use case scenario. The selected AI pipeline, Decomposition-Residuals Deep Neural Network (DR-DNN), was created by Theodorakos et al. (2022) for electricity demand forecasting. In the software demo, users can configure, visualize, and comprehend the AI pipeline with integrated expert guidance to aid their understanding. As a prototype, this demo serves as a practical illustration of the framework's potential usability and accessibility, rather than a fully developed product for large-scale deployment.

Our research objective is not only to propose this framework but also to create a system that can be easily accessible to a larger audience. By offering an alternative method for researchers to present their work, we aim to bridge the gap between complex AI systems and broader, less-experienced audiences. This approach ensures that our methodology is practical and understandable for non-technical users, fostering wider adoption and comprehension of advanced AI technologies.

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. Fig. 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.

Fig. 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. 2A). The UI serves as the main communication channel between the user and the system, as described in Section 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 5.

Since the data is provided by an external source, it is directed into the system through the Data Management subsystem (Fig. 2B), bypassing the UI. The Data Management subsystem (Fig. 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. 2H), which contains the datasets. The second is the User's Database (Fig. 2I), holding all information related to users and project sessions, such as pipeline configurations and project descriptions. The third is the DSS Database (Fig. 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. 2E), is where the user provides initial details to start a new project. This information is managed by the Data Management subsystem (Fig. 2B) and saved in the database (Fig. 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. 2G) then triggers the Assessment Flow (Fig. 2F). The Analysis Flow manages the user's actions during pipeline configuration, while the Assessment Flow (Fig. 2F) delivers suggestions based on expert knowledge and saves the user's feedback. The Analysis Flow (Fig. 2G) is linked to the Model Management subsystem (Fig. 2D), which, unlike traditional frameworks described in Section 2.3, consists of two parts: the Topological (Fig. 2D.1) and Hyperparameter (Fig. 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 logs, hidden size, learning rate, and others. The Model Management subsystem (Fig. 2D) is also responsible for saving and running the configured pipeline.

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

The Assessment Flow (Fig. 2F) is directly linked to the Knowledge Management subsystem (Fig. 2C), which includes the Expert Analyzer (Fig. 2C.1). The Expert Analyzer accesses stored expert knowledge and retrieves relevant information to make suggestions, which are then sent to the UI (Fig. 2A) via the Suggestion Flow (Fig. 2F.1). Additionally, the Knowledge Management subsystem (Fig. 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. 2K) transfers information about parameter settings and pipeline structure from the Data Management subsystem (Fig. 2B) to the Model Management subsystem (Fig. 2D). The User's Instruction Link (Fig. 2L) conveys user choices from the Model Management subsystem (Fig. 2D) to the Knowledge Management subsystem (Fig. 2C) for tailored suggestions. During

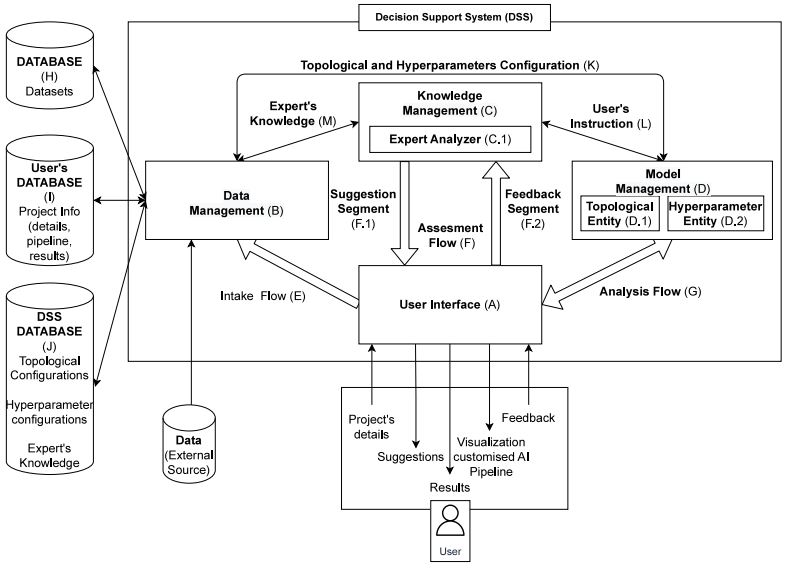


Fig. 2. The DSS4EX framework. DSS4EX framework represents a Decision Support System (DSS) tailored for systems that aid users in exploring Artificial Intelligence pipelines. It comprises the classical subsystem of a DSS, but the Knowledge Management subsystem and the Model Management subsystem were reinforced with more details specific to the scope. The interaction between the interface and each subsystem is described by the flow of information interchanged.

this process, the connection (Fig. 2M) between the Data Management subsystem (Fig. 2B) and the Knowledge Management subsystem (Fig. 2C) retrieves expert knowledge from the database.

Having discussed what the subsystems and the links are and their function, Section 5 discusses how to use the framework in a practical application using a deep learning pipeline on time series data.

5. Application of the framework: DSS4EX for the decomposition residuals deep neural network pipeline

This section describes a software demo that validates the effectiveness of the DSS4EX framework. It is divided into four subsections: the context, the dataset, the technologies used to develop the application, and the different parts that constitute the software demo.

5.1. 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 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 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) and the real-life problem is to predict the electricity demand for the following day.

5.2. The artificial intelligence pipeline for time series forecasting

The target problem described in Section 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 (Lim & Zohren, 2021; Meisenbacher et al., 2022). Many pipelines are available in the literature (Lim & Zohren, 2021; Meisenbacher et al., 2022), and in this paper, the architecture described in Theodorakos et al. (2022) 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 Fig. 4, 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 h 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.
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-h 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.

5.3. The dataset

The dataset used in the software demo is the same one as the paper introducing the DR-DNN Pipeline (Theodorakos et al., 2022). It is a

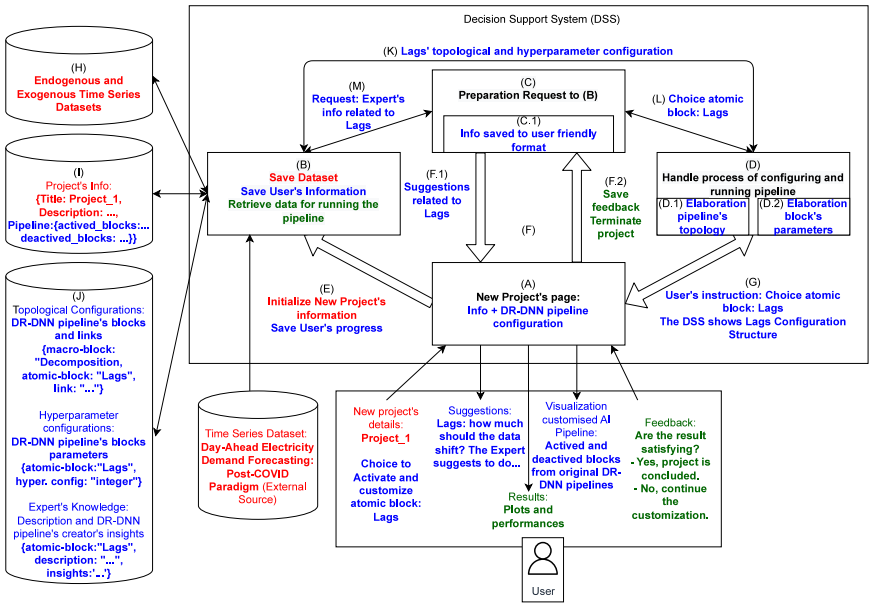


Fig. 3. Example of interaction when mapping the DSS4EX to the Decomposition Residuals Deep Neural Network (DR-DNN) use case. In the image, the information in red is the flow related to the initialization of a new project; in blue is the information flow of the selection of the Lags blocks; and in green, the flow related to the training and visualization of the results. The user creates a new project through the interface (A). This information navigates to the Data Management subsystem (B) using the Intake Flow (E) and is saved into the database (I). When the user wants to visualize the pipeline, the Interface (A) activates the Analysis Flow (G) so it can request the topological configuration from the database (J) to the Model Management subsystem (D), in particular to the Topological Entity (D.1). The image shows as an example the information flow inside the DSS when the user wants to modify the configuration of the Lags block. The DSS uses the communication channel Analysis Flow (G) to request the details of the parameters to the Hyperparameter Entity (D.2). The Model Management subsystem (D) additionally sends the user's instruction (L) to the Knowledge Management subsystem (C) and its Expert Analyzer (C.1) elaborate a pertinent suggestion that is sent to the interface through Assessment Flow (F). At the end, the user sends the trigger to the Model Management subsystem (D) and receives results and insights from it and the Knowledge Management subsystem (C). Ultimately, the pipeline configuration is applied to the time series dataset (specifically "Day-Ahead Electricity Demand Forecasting: Post-COVID Paradigm".)

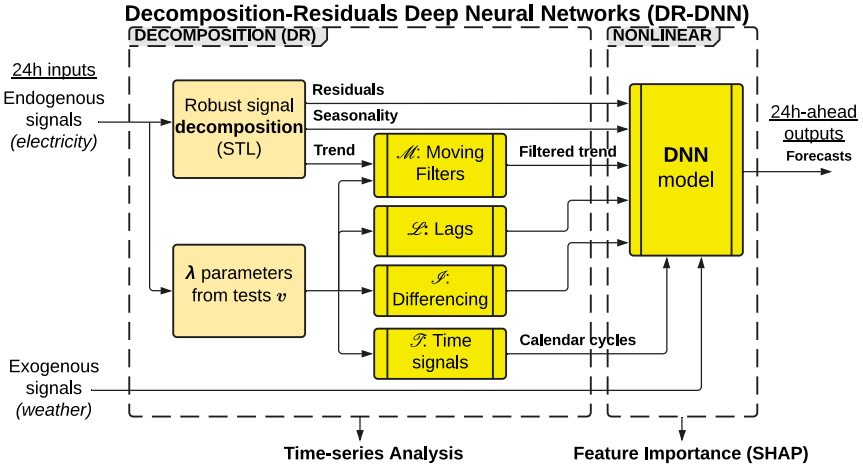


Fig. 4. The Decomposition Residuals Deep Neural Network (DR-DNN) pipeline proposed by Theodorakos et al. (2022). It is a two-layer pipeline. The first part, the Decomposition Residuals layer, extracts information (trend, seasonalities, and residuals) from the endogenous signal (the variable to be forecasted), 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).

Table 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" (Farrokhabadi et al., 2022). 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".

Time	Load (kW)	Pressure_kpa	Cloud Cover (%)	Humidity (%)	Temperature (C)	Wind Direction (deg)	Wind Speed (kmh)
2017-03-18 08:00:00	1 060 602	1011,869	44	77,41667	15,83188622	268,375	5,28125
2017-03-18 09:00:00	1 141 663	1013,41	44	74,33333	19,36938622	289,375	7,13375
2017-03-18 10:00:00	1 195 235	1013,54	44	74,33333	21,59438622	285,625	14,72375
2017-03-18 11:00:00	1 208 175	1013,086	44	76,25	22,38688622	279,375	16,6312
2017-03-18 12:00:00	1 216 409	1013,589	44	77,91667	21,39188622	270,75	13,36375
2017-03-18 13:00:00	1 211 772	1013,695	44	76	22,12938622	282	16,0625

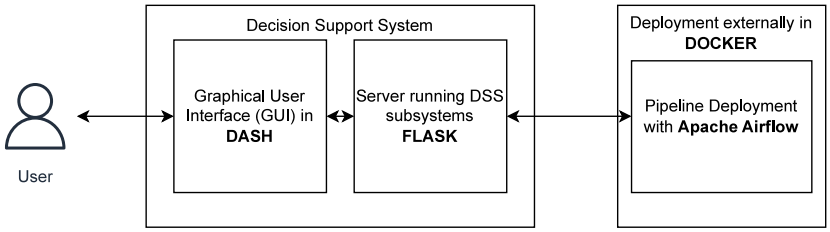


Fig. 5. Interaction of DSS4EX Technologies This figure illustrates the technological interactions within DSS4EX. The user interacts with the Decision Support System (DSS) through a Graphical User Interface (GUI) developed in Dash. The GUI retrieves information processed by the server, which hosts DSS 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.

public time series dataset titled "Day-Ahead Electricity Demand Forecasting: Post-COVID Paradigm".[7] 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 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.

5.4. The technologies

The software demo, illustrated in this section, is a web application written in Python, leveraging the web application framework Dash[8] and integrating Apache Airflow[9] for computationally intensive tasks. Python was chosen due to its extensive use in both web development and data manipulation, along with its vast ecosystem of libraries. Additionally, the pipeline's source code[10] is implemented in Python, making it an ideal choice for compatibility and ease of integration. Dash offers high flexibility and control over interface interactivity, setting it apart from competing frameworks.

In more detail, as represented in Fig. 5, the web interface is developed using Dash, while the web server-side application is implemented in Flask,[11] which handles the data management, knowledge management, and model management subsystems. Flask, as a lightweight micro-framework, enables fast development and low computational overhead. Although Flask is not designed for production-grade scalability, it was selected for DSS4EX due to its suitability for prototyping. The user-configured training process is deployed on Apache Airflow and runs on an additional server that hosts a Docker[12] container with PyTorch[13] installed to support deep learning models.

Apache Airflow was chosen for its robust workflow management capabilities, which streamline the orchestration of complex computational tasks. Docker was used to simulate an external service setup for model training and receiving results. This configuration enables the efficient organization and execution of the DSS4EX pipeline, ensuring that training tasks are managed independently and effectively within a dedicated server environment. Appendix B provides a detailed list of the packages used during development.

5.5. Software demo description

As mentioned, the software demo shows a practical usage of the DSS4EX framework with the DR-DNN pipeline for electricity demand forecasting. Fig. 3 shows the framework mapped to this use case. In the image, the information flow of creating a new project, the interaction for modifying the Lags block, and the model training with the consequent visualization of results are represented in the colors red, blue, and green, respectively. It illustrates an example of how each component works when the user creates a new project and selects the block Lags for configuration. Figs. 7 and 8 illustrate the software demo's interface through screenshots. The same "Lags" example used to map the framework (Fig. 3) was implemented to show the interface when necessary. Instead, Fig. 6 shows the embedded video, which demonstrates the functionalities discussed in this Section.

[7] https://ieee-dataport.org/competitions/day-ahead-electricity-demand-forecasting-post-covid-paradigm.
[8] https://dash.plotly.com/.
[9] https://airflow.apache.org/.
[10] https://github.com/temp3rr0r/DR-DNN.
[11] https://flask.palletsprojects.com/en/stable/.
[12] https://www.docker.com/.
[13] https://pytorch.org/.

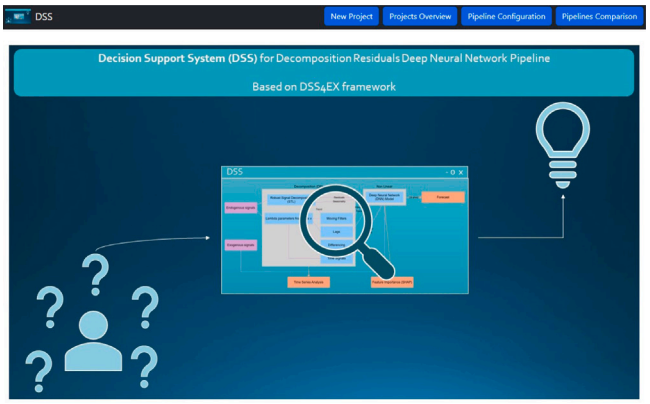


Fig. 6. Software Demo Video. The video illustrates the functionalities of the software demo designed using the DSS4EX framework.

5.5.1. New project

The first user–system interaction is the initialization of a new project. The interface (Fig. 3A) shows the New Project Page (Fig. 7(a)) 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".

After saving this information (Fig. 3A), the interface triggers the Intake Flow (Fig. 3E) that sends it to the Data Management subsystem (Fig. 3B). This stores the title and the description in the User's Database (Fig. 3I) as a new project.

5.5.2. Pipeline visualization

After the new project information is saved, the interface (Fig. 3A) shows the list of projects (Fig. 7(b)), 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), and it is the only one available. As described in Section 5.2 and shown in Fig. 4, 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 h of data (one day), then tries to predict the next 24 h, 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.

5.5.3. Pipeline configuration

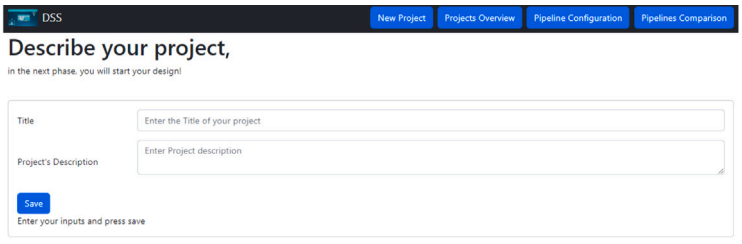
After initializing the project and visualizing the pipeline, the user can access the Configuration Page (Fig. 7(c)), where it is possible to read the description of each block, receive the designer's suggestion, and decide about configuration parameters. The description and the required configuration are retrieved by the Model Management subsystem (Fig. 3D) through the Data Management subsystem (Fig. 3B), which takes this information from the database (Fig. 3J). In detail, the Topological Entity (Fig. 3D.1) will retrieve and collect the structure of the pipeline, instead of the Hyperparameter Entity (Fig. 3D.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.4 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. 3C) stored in the database (Fig. 3J), as well through the Data Management subsystem (Fig. 3B). The suggestions are based on the expert's insights (Fig. 3J), which are static texts, and they are sent to the User Interface (Fig. 3A) by the Suggestion Segment (Fig. 3F.1) for this software demo. Fig. 7(c) 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. 3D) through the Data Management subsystem (Fig. 3B) into the database (Fig. 3J). Table 2 and Fig. 9(a) 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 and Fig. 10(a) depict the same information but for the project "Pipeline_DR-DNN".

5.5.4. Pipeline training

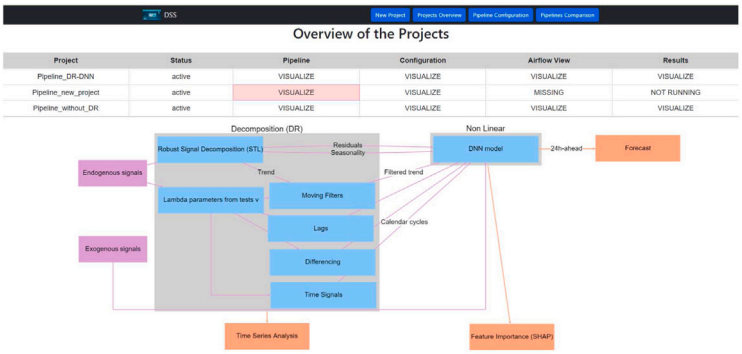
After configuring all the blocks, the user triggers the Analysis Flow (Fig. 3G) that activates the Model Management subsystem (Fig. 3D) to run the pipeline through the interface (Fig. 3A). As in the case of this software demo, the Model Management subsystem (Fig. 3D) can be connected to a separate service that handles the actual computation in order to improve efficiency and reduce execution time. This means that the Model Management subsystem (Fig. 3D) needs to retrieve the configuration and prepare it for the external service. It also recovers the dataset from the database (Fig. 3H) through the Data Management subsystem (Fig. 3B). 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. 3A), as shown in Fig. 8(a). In the software demo, the external service starts running the new pipeline in less than 4 min. The time of running a pipeline with only active the minimum required blocks (the Endogenous System block, the Exogenous System block, and the DNN Model) is less than 3 min. This is approximately the time that the model needs when all the blocks are configured. The reason lies in the fact that the Decomposition Residuals phase is not computationally time-consuming. The training of the DNN model is the most demanding phase. Given its characteristics, the computation time for training can fluctuate, and each execution can yield different outcomes due to varying random initializations. In the previously described example, "Pipeline_without_DR" was trained in 1 min and 55 s instead of "Pipeline_DR-DNN" in 2 min and 17 s.

5.5.5. Pipeline evaluation and results

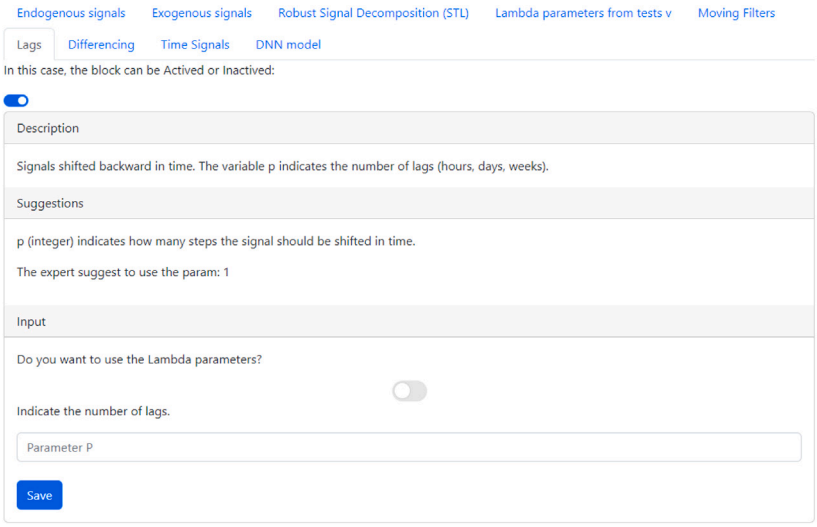
After the pipeline is trained, the Model Management subsystem (Fig. 3D) collects the results and saves them through the Data Management subsystem (Fig. 3B) into the database (Fig. 3I). When the user requests the results, the Model Management subsystem (Fig. 3D) retrieves and sends them to the User Interface (Fig. 3A). An example of how the Result page looks like is shown in Fig. 8(b). 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



(a) New Project Page



(b) Overview Projects Page



(c) Configuration Page

Fig. 7. Screenshots taken from the software demo based on the DSS4EX framework. Image (a) shows the page dedicated to initiating a new project. Image (b) displays the listed projects and the pipeline visualization. Image (c) depicts how the user can modify the parameters of each pipeline's block.

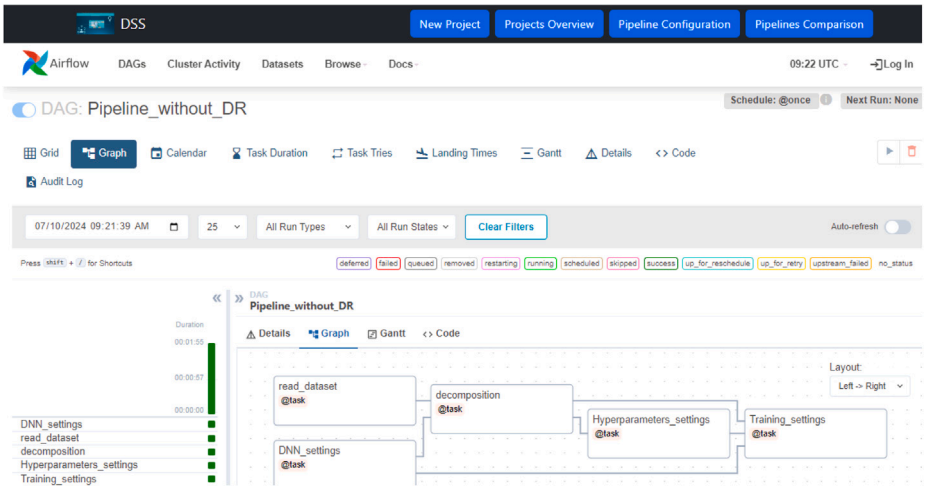
Table 2
Values chosen to run the experiments that show the functionalities of the software demo.

Projects	Train/Val./Test	Normalization	Exogenous	STL/parameter(period)	λ Test	Moving Filter	Lags	Differencing
Pipeline_without_DR	70/20/10	YES	All	Skipped	Skipped	Skipped	Skipped	Skipped
Pipeline_DR-DNN	70/20/10	YES	All	Yes/7	Skipped	2	1	0

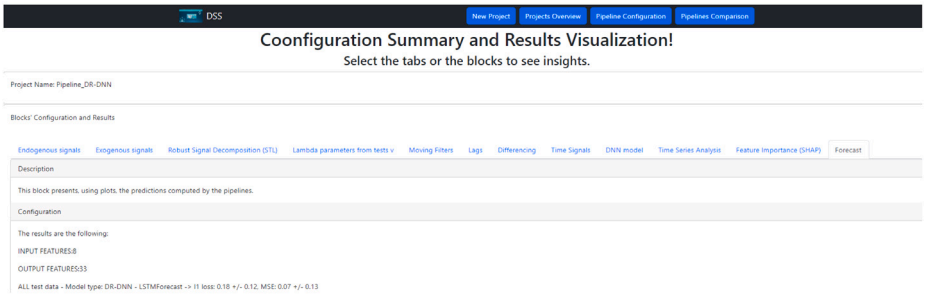
Table 3
Hyperparameter values used to train the non-linear (DNN) model for both the example pipelines.

Hidden size	Output size	Batch size	Learning rate	Epochs
64	24	32	0.001	20

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 Figs. 9(c) and 10(c) 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. 9(c)),

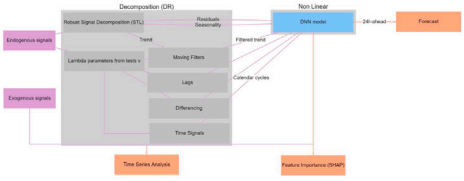


(a) Training View Page

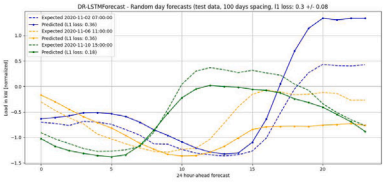


(b) Configuration Summary and Results Page

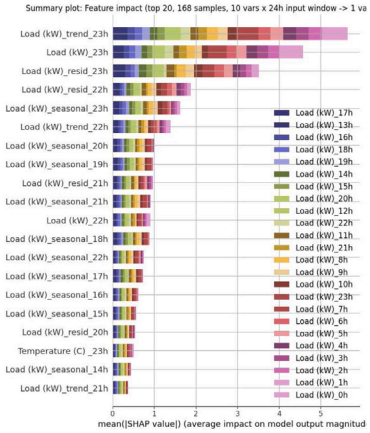
Fig. 8. Screenshots taken from the software demo based on the DSS4EX framework. Image (a) shows the view of the external service that runs and trains the pipeline. Image (b) presents the page displaying the configuration of the pipeline as well as the results (plots, error metrics, etc.).



(a) Pipeline_without_DR overview after configuration.

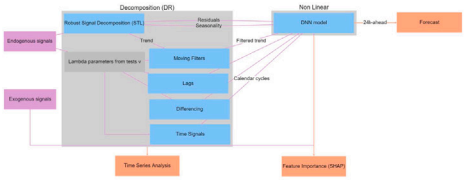


(b) Plots of the predictions of three days compared with their expected values.

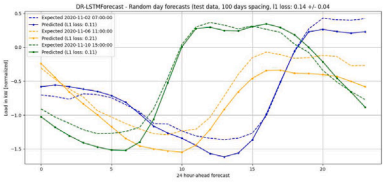


(c) Most significant features plotted using SHAP.

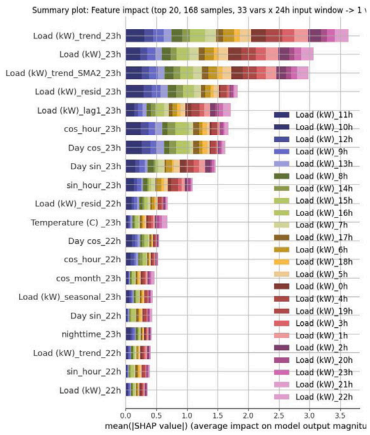
Fig. 9. Overview of the pipeline, results, and SHAP plots of the project "Pipeline_without_DR". Image (a) shows the pipeline after configuration, with gray blocks representing deactivated components related to the DR layer. Image (b) displays the model's predictions alongside expected values for three different days. Image (c) 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) Plots of the predictions of three days compared with their expected values.



(c) Most significant features plotted using SHAP.

Fig. 10. Overview of the pipeline, results and SHAP plots of the project "Pipeline_DR-DNN". Image (a) presents the pipeline after configuration. The gray 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 different days. In Image (c), the 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.

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. 10(c) 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. Figs. 9(b) and 10(b) 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 Theodorakos et al.'s results (Theodorakos et al., 2022), 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. 7(b)), triggering the Feedback Segment (Fig. 3F.2) and indicating to the system and the Knowledge Management subsystem (Fig. 3C) that the project is finished.

5.5.6. Pipelines comparison

The Knowledge Management subsystem (Fig. 3C) and the Model Management subsystem (Fig. 3D) 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. 3C) and the Model Management subsystem (Fig. 3D) work to retrieve and compute the necessary information using results and insights fetched by the Data Management subsystem (Fig. 3B) from the database (Fig. 3I) and showed through the User Interface (Fig. 3A). Figs. 11 and 12 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 expected and actual values. To complete the comparison, some metrics are calculated for each project on the selected time slots. In particular, Figs. 11 and 12 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".

Fig. 11 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. Fig. 12 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. 3C) provides static descriptions to help the user understand the plots and the metrics.

6. Results and discussions

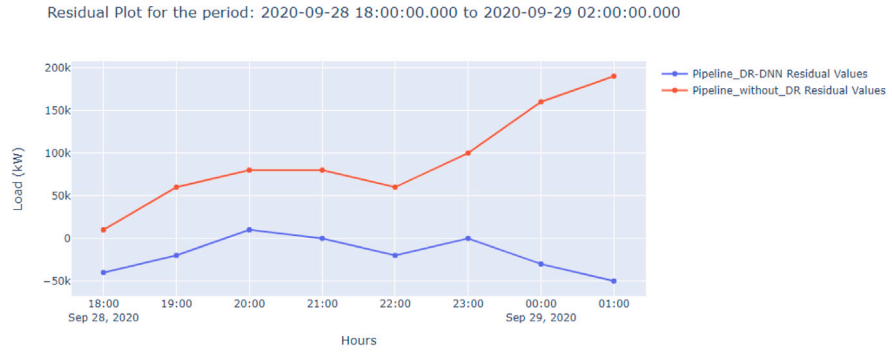
This study presents DSS4EX, a Decision Support System (DSS) framework designed to simplify the exploration, configuration, and comprehension of complex AI pipelines by both expert and non-expert users. To demonstrate DSS4EX's functionality, a prototype was developed as a proof-of-concept. This prototype 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 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 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). The interface, shown in Fig. 6, showed to be effective in enabling users to interact directly with the pipeline, offering responsive insights into its components and configurations (Section 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 5.5).



Fig. 11. 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

Fig. 12. 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.

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. 7(c)). 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. This feedback suggests the framework's potential and adaptability for both expert and non-expert users. 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 7.

7. Conclusions and future works

This paper 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 (Theodorakos et al., 2022).

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.

Table A.4
Name and descriptions of each block in the pipeline. This information is shown to the user during the configuration phase.

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.
λ parameters from tests ν	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.
Forecast	This block presents, using plots, the predictions computed by the pipelines.

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

Table B.5
Name, version and descriptions of the Python packages. Packages Used in DSS4EX and Their Descriptions,.

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/
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/
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/
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/

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.

CRediT authorship contribution statement

Giulia Rinaldi: Conceptualization, Methodology, Software, Formal analysis, Investigation, Resources, Writing – original draft, Writing – review & editing, Visualization. **Konstantinos Theodorakos:** Validation, Investigation, Resources, Data curation, Writing – review & editing. **Fernando Crema Garcia:** Software, Validation. **Oscar Mauricio Agudelo:** Writing – review & editing Supervision, Project administration. **Bart De Moor:** Supervision, Funding acquisition.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the author(s) used ChatGPT4 and Grammarly in order to improve language and readability. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

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.

Appendix A. Block descriptions

This section provides an illustration of the blocks utilized in the Decomposition Residual - Deep Neural Networks (DR-DNN) pipeline, as applied in the context described in this paper (see Table A.4).

Appendix B. Packages requirements

This section outlines the packages and their respective versions utilized in developing the DSS4EX prototype presented in this paper. These specific versions were selected to ensure compatibility, stability, and reproducibility of the results, supporting the integrity of the framework (see Table B.5).

Data availability

I have shared the data link and code in the text.

References

Amershi, S., Arksey, N., Carenini, G., Conati, C., Mackworth, A., Maclaren, H., & Poole, D. (2005). 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* (pp. 178–182). New York, NY, USA: Association for Computing Machinery.

Beets, B., Newman, T., Howell, E., Bao, L., & Yang, S. (2023). Surveying public perceptions of artificial intelligence in health care in the United States: Systematic review. *Journal of Medical Internet Research*, *25*, Article e40337.

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

Brauner, P., Hick, A., Philipsen, R., & Ziefle, M. (2023). 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*, URL https://www.frontiersin.org/journals/computer-science/articles/10.3389/fcomp.2023.1113903.

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

Cave, S., Coughlan, K., & Dihal, K. (2019). "Scary robots": Examining public responses to AI. In *Proceedings of the 2019 AAAI/ACM conference on AI, ethics, and society* (pp. 331–337). New York, NY, USA: Association for Computing Machinery.

Clement, T., Kemmerzell, N., Abdelaal, M., & Amberg, M. (2023). XAIR: A systematic metareview of explainable AI (XAI) aligned to the software development process. *Machine Learning and Knowledge Extraction*, *5*(1), 78–108, URL https://www.mdpi.com/2504-4990/5/1/6.

De Angelis, L., Baglivo, F., Arzilli, G., Privitera, G. P., Ferragina, P., Tozzi, A. E., & Rizzo, C. (2023). ChatGPT and the rise of large language models: the new AI-driven infodemic threat in public health. *Frontiers in Public Health*.

Dellermann, D., Lipusch, N., Ebel, P., & Leimeister, J. M. (2019). Design principles for a hybrid intelligence decision support system for business model validation. *Electronic Markets*, *29*, 423–441, URL https://link.springer.com/article/10.1007/s12525-018-0309-2.

Elragal, A., & Elgendy, N. (2024). Design principles supporting data-driven decisions platforms. *Research Square*, URL https://www.researchsquare.com/article/rs-4240652/v1.

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

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

Feurer, M., Eggensperger, K., Falkner, S., Lindauer, M., & Hutter, F. (2022). Auto-sklearn 2.0: Hands-free AutoML via meta-learning. *Journal of Machine Learning Research*, *23*(261), 1–61, URL http://jmlr.org/papers/v23/21-0992.html.

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

Ignatiev, A. (2020). Towards trustable explainable AI. In *IJCAI* (pp. 5154–5158).

Jarrett, D., Yoon, J., Bica, I., Qian, Z., Ercole, A., & van der Schaar, M. (2021). Clairvoyance: A pipeline toolkit for medical time series. In *International conference on learning representations*.

Joey, G. F., & Baldelovar, M. (2022). Decision support system: Overview, different types and elements. *Echnoarete Transactions on Intelligent Data Mining and Knowledge Discovery*.

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

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., & Stumpf, S. (2024). Explainable artificial intelligence (XAI) 2.0: A manifesto of open challenges and interdisciplinary research directions. *Information Fusion*, *106*, Article 102301, URL https://www.sciencedirect.com/science/article/pii/S1566253524000794.

Maarif, M. R., Saleh, A. R., Habibi, M., Fitriyani, N. L., & Syafrudin, M. (2023). Energy usage forecasting model based on long short-term memory (LSTM) and explainable artificial intelligence (XAI). *Information*, *14*(5), URL https://www.mdpi.com/2078-2489/14/5/265.

Meisenbacher, S., Turowski, M., Phipps, K., Rätz, M., Müller, D., Hagenmeyer, V., & Mikut, R. (2022). Review of automated time series forecasting pipelines. CoRR, abs/2202.01712.

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

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

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

Pelekis, S., Pountridis, T., Kormpakis, G., Lampropoulos, G., Karakolis, E., Mouzakitis, S., & Askounis, D. (2024). DeepTSF: Codeless machine learning operations for time series forecasting. *SoftwareX*, *27*, Article 101758, URL https://www.sciencedirect.com/science/article/pii/S2352711024001298.

Rupnik, R., Kukar, M., Vračar, P., Košir, D., Pevec, D., & Bosnić, Z. (2019). AgroDSS: A decision support system for agriculture and farming. *Computers and Electronics in Agriculture*, *161*, 260–271.

Salehin, I., Islam, M. S., Saha, P., Noman, S., Tuni, A., Hasan, M. M., & Baten, M. A. (2024). AutoML: A systematic review on automated machine learning with neural architecture search. *Journal of Information and Intelligence*, *2*(1), 52–81, URL https://www.sciencedirect.com/science/article/pii/S2949715923000604.

Shchur, O., Turkmen, C., Erickson, N., Shen, H., Shirkov, A., Hu, T., & Wang, Y. (2023). AutoGluon-TimeSeries: Automl for probabilistic time series forecasting. arXiv:2308.05566. URL https://arxiv.org/abs/2308.05566.

Sjöström, J., Dryselius, P., Nygren, J., Nair, M., Soliman, A., & Lundgren, L. E. (2024). Design principles for machine learning based clinical decision support systems: A design science study. In *Lecture notes in computer science*: *vol. 14621*, *Design science research for a resilient future* (pp. 109–122). Springer, URL https://link.springer.com/chapter/10.1007/978-3-031-61175-9_8.

Sprague, R. H. (1980). A framework for the development of decision support systems. *MIS Quarterly*, *4*(4), 1–26.

Stamenković, S., Jovanović, N., Vasović, B., Cvjetković, M., & Jovanović, Z. (2023). Software tools for learning artificial intelligence algorithms. *Artificial Intelligence Review*, *56*(9), 10297–10326.

Territo, K., & Romagnoli, J. (2024). FASTMAN-JMP: All-in-one tool for data mining and model building. In F. Manenti, & G. V. Reklaitis (Eds.), *Computer aided chemical engineering*: *vol. 53*, *34th European symposium on computer aided process engineering / 15th international symposium on process systems engineering* (pp. 3421–3426). Elsevier, URL https://www.sciencedirect.com/science/article/pii/B9780443288241505718.

Theodorakos, K., Agudelo, O. M., Espinoza, M., & De Moor, B. (2022). Decomposition-residuals neural networks: Hybrid system identification applied to electricity demand forecasting. *IEEE Open Access Journal of Power and Energy*, *9*, 241–253.

Turban, E., Sharda, R., & Delen, D. (2010). *Decision support and business intelligence systems* (9th ed.). Prentice Hall Press.

Wang, Y.-C., Chen, T.-C., & Chiu, M.-C. (2023). An improved explainable artificial intelligence tool in healthcare for hospital recommendation. *Healthcare Analytics*, *3*, Article 100147.

Westergaard, G., Erden, U., Mateo, O. A., Lampo, S. M., Akinci, T. C., & Topsakal, O. (2024). Time series forecasting utilizing automated machine learning (AutoML): A comparative analysis study on diverse datasets. *Information*, *15*(1), URL https://www.mdpi.com/2078-2489/15/1/39.

Wu, J., Wang, H., Ni, C., Zhang, C., & Lu, W. (2024). Data pipeline training: Integrating AutoML to optimize the data flow of machine learning models. arXiv:2402.12916. URL https://arxiv.org/abs/2402.12916.

Zhang, Y., Sun, Q., Qi, D., Liu, J., Ma, R., & Petrosian, O. (2024). *Shaptime: A general XAI approach for explainable time series forecasting* (pp. 659–673). Springer.

Zhou, C., Kuang, D., Liu, J., Yang, H., Zhang, Z., Mackworth, A., & Poole, D. (2020). AISpace2: An interactive visualization tool for learning and teaching artificial intelligence. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(09), 13436–13443, URL https://ojs.aaai.org/index.php/AAAI/article/view/7068.

Zielinska, K. (2024). 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*.