AI Explainability Methods in Digital Twins: A Model and a Use Case

Tim Kreuzer[1 [0000−0002−0813−9555]], Panagiotis Papapetrou[1 [0000−0002−4632−4815]], and Jelena Zdravkovic[1 [0000−0002−0870−0330]]

Stockholm University, 16455 Kista, Sweden
{tim.kreuzer,panagiotis,jelenaz}@dsv.su.se

Abstract. Digital twin systems can benefit from the integration of artificial intelligence (AI) algorithms for providing for example some predictive capabilities or supporting internal decision-making. As AI algorithms are often opaque, it becomes necessary to explain their decisions to a human operator working with the digital twin. In this study, we investigate the integration of explainable AI techniques with digital twins, which we termed XAI-DT system. We define the concept of XAI-DT system and provide a use case in smart buildings, where explainable AI is used to forecast CO_2 concentration. Further, we present a core architectural model for our digital twin, outlining its interaction with the smart building and its internal processing. We evaluate five AI algorithms and compare their explainability for the operator and the entire digital twin model based on standard explainability properties from the literature.

Keywords: Digital Twin · Explainable AI · System Analysis · Forecasting

1 Introduction

A digital twin (DT) is a virtual replica of a physical system, operating in real time on the basis of a bidirectional data stream. The concept has recently gained traction in research [26, 31, 33], and is increasingly applied in industrial scenarios [4, 38], mainly in the domain of manufacturing [34]. A DT mirrors the behavior of its physical counterpart while providing further capabilities benefitting the end-users of the system. This can be achieved by integrating artificial intelligence (AI) methods with a DT, allowing thus performing complex predictive and analysis functions based on the data it processes [17]. For instance, by employing AI algorithms, a DT can forecast the energy demand of a power plant [36] or regulate the heating and ventilation of a smart building using a forecast of temperature and CO_2 concentration [6]. This demonstrates that a digital twin can benefit from AI-based forecasts, which can be made with various machine learning techniques.

Digital twins frequently work in conjunction with human operators [9], who receive feedback from the DT system and can act upon it. When integrating AI with a DT, it is crucial to use explainable AI methods [16], as they can help the

operator of the system to understand the output of the AI model and the mechanisms that lead to this output. As an example, in the case of a smart building, a facility manager would want to know how a machine learning model came to the conclusion that CO_2 concentration would increase by a certain amount in the next hour. In this scenario, the DT needs to work with explainable AI techniques that are able to both make accurate forecasts and provide explanations for them. As the explainability of an AI algorithm is highly context-specific [2], it depends on the application domain and the problem at hand. With a smart building, for example, an explanation could be based on a cyclic, reoccurring pattern where CO_2 concentration usually increases at 9:00 due to a regular meeting at this time. Explanations can be more complex when more variables are involved: CO_2 concentration might be forecasted to rise due to a measured increase in the number of people in a meeting room, while the ventilation system is scheduled to turn on only an hour later. Explainable AI can help provide such explanations to operators of a digital twin.

Contributions. This paper investigates the problem of integrating explainable AI techniques with digital twins. For this, the addressed research question is: *How can machine learning explainability methods be integrated with digital twins?* More concretely, we formally define the terms *Digital Twin*, *AI-DT system*, and introduce the concept of *XAI-DT system*, combining explainable AI techniques with digital twins. We present a real use case where explainable forecasting methods are used within a DT of a smart building. Moreover, we introduce a core architectural model of a DT for our use case, connecting the proposed definitions with a realistic scenario. Finally, we validate the outlined DT by comparing the performance and explainability of multiple AI methods based on real-world data. We systematically assess explainability based on a set of well-known explainability properties from the literature [23].

2 Background

Artificial intelligence has been employed for a multitude of tasks in digital twins [17] such as optimization, classification, forecasting and outlier (i.e. anomaly) detection; as well as in many domains [27] including manufacturing, medical and transportation. Wang et al. [35] have, for example, introduced a traffic digital twin, replicating drivers, vehicles, and traffic, where AI is used to classify driver types. Li et al. [18] have proposed a system for computing task offloading using reinforcement learning in conjunction with digital twins of unmanned aerial vehicles and mobile terminals. Matulis et al. [22] have used reinforcement learning for movement optimization of a robotic arm, training the arm in a digital twin-based environment.

Explainable AI (XAI) is a field that has recently experienced an increase in interest [25]. Explainability in machine learning can be achieved with *inherently explainable models* that are explainable due to their internal mechanisms without any postprocessing. Alternatively, model-agnostic *post hoc methods* can be used to explain black-box models that are not inherently explainable. Based on the

work by Burkart and Huber [5], interpretable models (white-box models) include linear models, decision trees, rule-based models, and Bayesian models. White-box models, however, usually come at a cost of reduced accuracy, flexibility, or usability [29].

In this work, we further follow the definitions of Molnar regarding explainability and the properties of explanations [23]: *Accuracy* shows how well an explanation predicts unseen data, which is only applicable when explanations are used to make predictions. Molnar describes *fidelity* as the property that measures how well an explanation reflects the prediction of the model, which is a key property of any explanation. *Consistency* is defined as the similarity of explanations for different models trained on the same task. *Stability* describes how much explanations of a single model differ for similar data points. Further, *comprehensibility* defines how well humans can understand an explanation. Without a certain degree of comprehensibility, explainable AI techniques are not practical. *Certainty* reflects the ability of an explanation to capture the model's uncertainty regarding its predictions. *Novelty* is connected to certainty; it is high when an explanation can determine that a given data point is novel. The *degree of importance* describes whether an explanation assigns importance to the features of the data. Lastly, *representativeness* characterizes whether an explanation covers a machine learning model as a whole or only an individual prediction.

For the task of time series forecasting, models commonly work on the basis of trend and seasonality and are, therefore, inherently explainable: the trend can be represented with a polynomial of a small degree, while the seasonality is a periodic function. Combining both can reveal the mechanism underlying the model's prediction. This approach is followed in N-BEATS [24] and D-Linear [37], two deep learning architectures for forecasting which decompose their output into trend and seasonality. Recently, transformer architectures have commonly been used for time series forecasting [20, 19, 39], which are black-box models that are not explainable. Other models, such as ARIMA [3], implement an autoregressive approach, which, despite being statistical in nature, are not inherently interpretable, as they are highly complex. Recent work has also explored the use of large language models for time series forecasting [10]. To explain black-box models' predictions, methods like SHAP [21], have been used in past research [8] to provide post hoc model explanations on a feature importance level. TS-MULE [30] is another method for post hoc explanations, assigning segment-based relevance scores to an input time series. Other explainability methods include counterfactuals [11], which suggest how the inputs should be changed to receive different outputs.

Researchers have investigated the multidisciplinary topic of XAI and digital twins, integrating the concept of explainable AI with a DT. Kapteyn et al. [15, 14] have used a decision tree, which is an inherently explainable model, for classification within a DT of an unmanned aerial vehicle. Suhail et al. [32] have presented a platform for a DT of a cyber-physical system, using SHAP to provide explainability. A similar approach was pursued by Kobayashi et al. [16], who

employ multiple post hoc approaches, including LIME and SHAP, to provide explainable remaining useful life estimation based on a deep neural network.

3 Explainable AI in Digital Twins

This section describes the integration of explainable AI techniques with digital twins. First, we give formal definitions of a digital twin, an AI-DT system, and an explainable AI-DT system to define the elementary components for the model presented in Fig. 1 . Further, we illustrate the definitions with a real-world use case based on the DT of a smart building, employing XAI methods to forecast CO_2 concentration. We provide a pseudo algorithm for the internal processing of the DT and showcase the process with an architectural model. Lastly, we compare multiple AI methods for use within the DT, comparing their performance and explainability on the forecasting task.

3.1 Definitions

Definition 1 (Digital Twin). *Let a* ***digital twin*** D *be a tuple, with* $D = (I, C, O)$*. A digital twin is a virtual replica of a* ***physical system****, represented by a set of variables* P*. The physical system is a necessary contextual element for the DT and provides a set of* ***input streams*** I*, which are connected to the digital twin, so that* $I \subseteq P$*. Based on* I*,* D *can accurately represent the physical system in its context. The* ***components*** C *within the digital twin can be of varied nature and provide internal processing capabilities. With the* ***output streams*** O*, the digital twin closes the feedback loop to the physical system, where each output stream is either directly or indirectly related to the physical system.*

To connect Definition 1 to a real-world example, we investigate a digital twin of a smart building, further referred to as D_s, where the smart building is the physical system in the context of the DT, which is represented by the set of variables P_s. The digital twin operates on the basis of sensors installed in the building, measuring time-dependent values such as temperature, CO_2 concentration, or ventilation. We further only consider a CO_2 concentration sensor as the basis for the digital twin. Internally, it processes the data, forecasting changes in CO_2 concentration for the next hour. The DT is connected to the ventilation control system, adapting ventilation based on the results of the internal processing. Further, a dashboard illustrates the current status of the system and the forecast for a human operator.

Definition 2 (Input Stream). *Given a digital twin* D*, an input stream* $i \in I$ *is a time-dependent vector of data points* i*, where* $i =< i_1, \ldots, i_t >$*. An input stream originates from the physical system in the context of* D *and is updated in real-time, with the time point* t *representing the most recent observation. Input streams can have different data types, such as real numbers, images, or text.*

To illustrate Definition 2 in the context of a smart building, the data stemming from an individual CO_2 concentration sensor contribute to an input stream. The sensor takes measurements at regular time intervals, resulting in a temporally ordered sequence of numeric values. When connecting this to a digital twin, the CO_2 concentration sensor acts as the source of the input stream i_C while the digital twin is processing the data.

Definition 3 (Component). *Given a digital twin D, a component $c \in C$ is a function that processes an input x, resulting in an output y, where $c : x \to y$. The input x of each component is based on a number of input streams and a number of outputs of other components. The output y can be one or more values of multiple data types. Components have distinctive functionality, where each component fulfills a specific role within D.*

To process the previously outlined input stream of CO_2 concentration data, a preprocessing component c_p is introduced. This component takes the input stream i_C as the only input so that $x = \{i_C\}$. Internally, c_p checks the input stream for missing data and imputes them accordingly. If outliers occur within i_C, the preprocessing component records the number of outliers but does not act on them. The output of c_p is the preprocessed input stream, referred to as p, and the number of found outliers k so that $y = \{p, k\}$.

Following the preprocessing component, a forecasting component c_f makes predictions based on the preprocessed input, where $x = \{p\}$. It employs a machine learning algorithm, such as N-BEATS [24], to make a forecast of CO_2 concentration for the next hour, referred to as f. The training phase of the machine learning algorithm is not described here as it is preliminary. With the forecast, the output of the component is defined as follows: $y = \{f\}$.

To visualize the internal activity of c_p, and c_f in the highlighted example, we describe a dashboard component c_d, taking the input $x = \{i_C, f, k\}$. Based on the observed values in i_C and the forecasted CO_2 concentration f, a line chart of past and predicted CO_2 concentrations can be plotted. Additionally, the number of detected outliers k, which is a possible indicator of sensor quality degradation, is indicated in the dashboard. Finally, we define the resulting dashboard, which is the output of this component, as $y = \{d\}$.

Lastly, a rule-based component c_r is introduced for the regulation of the ventilation control system based on the forecasted CO_2 concentration. Its input $x = \{i_C, k, f\}$ is processed, leading to an output $y = \{r\}$ representing the control sequence sent to the ventilation control system.

Definition 4 (Output Stream). *Given a digital twin D, an output stream $o \in O$ is a time-dependent vector of outputs o, where each output is based on I and C. An output stream directly or indirectly affects the physical system P in the context of the digital twin. Similar to input streams, the data type of an output stream can vary depending on the application case.*

With the input stream i_C and the processing within the components c_p, c_f, and c_d, the dashboard d is a resulting output. The dashboard is time-dependent,

as it is based on the streamed CO_2 concentration data, which change over time. This output stream o_d connects the digital twin to a human operator who can act on the information presented in the dashboard. o_d is indirectly connected to the physical system P through the operator.

A second output stream o_r directly connects the digital twin to the physical system, streaming the control sequence r, resulting from c_r, to the physical ventilation control system. This closes the feedback loop between D_s and P_s, allowing the twin to influence the smart building based on its internal processing and decision-making.

Finally, given the physical system P_s of a smart building, the outlined digital twin D_s can be described as follows:

$$D_s = (\{i_C\}, \{c_p, c_f, c_d, c_r\}, \{o_d, o_r\}) \tag{1}$$

The digital twin of a smart building outlined in this section employs AI algorithms to forecast CO_2 concentration. This is an example where artificial intelligence is integrated with a digital twin, forming an *AI-DT system*. In the literature [1, 12], similar terms to AI-DT system have been used to characterize the concept of integrating AI with a digital twin, however, the term has not been defined in the past, so we give a formal definition here:

Definition 5 (AI-DT system). *We define an* ***AI-DT system*** *as a type of digital twin where artificial intelligence algorithms are integrated with the DT so that $\exists c \in C$, where c is a component of the digital twin that employs artificial intelligence to compute its outputs.*

Definition 6 (XAI-DT system). *Consider an AI-DT system with an AI component c. When c is based on inherently explainable AI techniques, we further refer to the AI-DT system as an* ***XAI-DT system****. When c is not inherently explainable and a second component c_x provides post hoc explanations based on c, the AI-DT system can also be considered an* ***XAI-DT system****.*

Expanding upon the smart building use case, D_s can be considered an AI-DT system, as its forecasting component c_f relies on a machine learning algorithm to make CO_2 concentration forecasts. D_s can further be considered an XAI-DT system when using an inherently explainable AI algorithm for the forecasting task. In this case, the output of c_f additionally includes inherent explanations e_i that are displayed by the dashboard component for the human operator. When using an opaque machine learning model that is not explainable, an additional post hoc explainability component needs to be added to make D_s an XAI-DT system. This explainability component c_x takes both the forecast and the machine learning model itself as inputs so that $x = \{f, M\}$ where M represents the machine learning model employed in c_f. Internally, c_x uses a post hoc explainability method such as SHAP, resulting in post hoc explanations e_p, illustrating the forecasting mechanism used to make the forecast so that $y = \{e_p\}$.

3.2 Use Case and Architectural Model

This section presents the overall architectural model of an XAI-DT system, instantiated for our use case in smart buildings. Fig. 1 shows the model while using the previously outlined definitions.

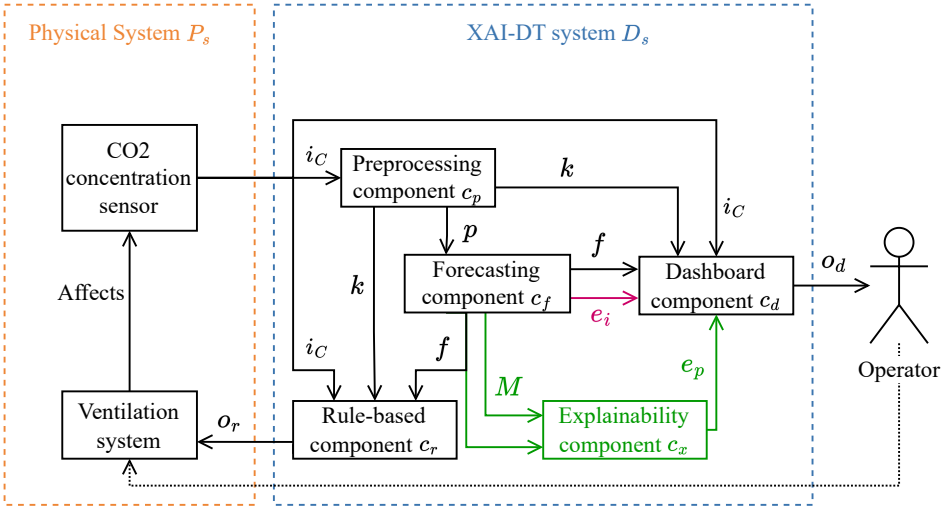


Fig. 1: Architectural model of D_s, representing a digital twin of a smart building P_s. This core model shows two possibilities for the integration of explainable AI: an inherently explainable algorithm providing inherent explanations e_i (pink) and a post hoc explainability component (green) providing post hoc explanations e_p for opaque models.

The physical system P_s represents the smart building in the context of the DT, containing a CO_2 concentration sensor that is affected by changes in the ventilation system and the environment in the smart building. Measurements from the CO_2 concentration sensor are streamed to the digital twin D_s, resulting in the input stream i_C that is used by the preprocessing, dashboard, and rule-based components. Internally, the components of the digital twin are connected, processing the CO_2 concentration input, making a forecast, and counting outliers, while c_f provides explanations to the dashboard. For opaque machine learning models, the figure also highlights the additional post hoc explainability component c_x (green) that provides explanations for the forecast made by the machine learning model and forwards them to the dashboard. With the explainability component and the forecasting component, D_s is an XAI-DT system. The output of the digital twin consists of two output streams: o_d, which streams a dashboard to the system's Operator, indirectly affecting the smart building, and secondly o_r, which sends a control sequence to the ventilation system, directly influencing P_s.

Algorithm 1 Data flow and processing of D_s with post hoc explainability

1: Receive i_C from the CO_2 concentration sensor in P_s
2: **Input:** CO_2 concentration data stream i_C.
3: $p, k \leftarrow c_p(i_C)$; Preprocess the CO_2 concentration data and count outliers.
4: $f, M \leftarrow c_f(p)$; Forecast CO_2 concentration for the next hour with an opaque AI algorithm.
5: $e_p \leftarrow c_x(f, M)$; Provide post hoc explanations for the AI algorithm and its forecasts.
6: $o_d \leftarrow c_d(i_C, f, k, e_p)$; Create a dashboard from measured CO_2 concentration, forecast, number of outliers, and model explanations.
7: $o_r \leftarrow c_r(f, k)$; Create a ventilation system control sequence based on forecast and number of outliers.
8: **Output:** Ventilation system control sequence o_r, system status dashboard o_d.
9: Send o_r to the ventilation system in P_s.

Algorithm 1 shows the data flow within D_s in a sequential order. The CO_2 concentration data is streamed from the sensor in P_s and used as an input stream i_C for the digital twin. Next, the input data is preprocessed and outliers are counted in c_p, resulting in p and k. Based on the preprocessed data, CO_2 concentration is forecasted for the next hour using an opaque AI algorithm, which does not provide explanations for its forecast. With the forecast f and the model M, the post hoc explainability component generates explanations e_p for the forecast, which are used in the dashboard. The dashboard additionally receives the input stream, the forecast, and the number of outliers, which are shown to the operator. Finally, the rule-based component c_r creates a control sequence for the ventilation system based on forecast and the number of outliers, resulting in the output stream o_r. The control sequence is forwarded to the ventilation system in P_s, closing the feedback loop between the digital twin and the physical system in its context.

3.3 Forecasting Methods

We conducted an evaluation of AI methods for the CO_2 concentration forecasting component c_f, comparing their accuracy. This evaluation is relevant to present because it aids in understanding which algorithms are effective for the proposed digital twin, which is an essential quality parameter in addition to explainability being needed for describing the mechanism leading to an output. We evaluate the forecasting algorithms shown in Table 1. Each algorithm providing some form of inherent explainability is additionally categorized as *Explainable*, which is evaluated in more detail in Sec. 3.4. All inherently explainable algorithms are categorized as explainable due to a decomposition-based forecast, which allows for a more fine-grained interpretation of the forecast.

N-BEATS and DEPTS were chosen as they provide decomposition-based explanations, while claiming high performance for the time series forecasting task. The non-stationary transformer was evaluated, as it is a non-explainable, transformer-based method, offering high performance as a black-box model.

Lastly, ARIMA was benchmarked to compare a statistical approach with the deep learning methods for time series forecasting.

Table 1: Algorithms evaluated

Algorithm name	Reference	Approach	Explainable
N-BEATS	[24]	Deep Learning	Yes
D-Linear	[37]	Deep Learning	Yes
Non-Stationary Transformer (NST)	[20]	Deep Learning	No
DEPTS	[7]	Deep Learning	Yes
ARIMA	[3]	Statistical	No

We are using historical CO_2 concentration data from a smart building dataset for the evaluation. The data consist of measurements of 106 sensors with a sampling frequency of 5 minutes that were gathered over a period of 10 days (for the purpose of this study, we have found this timeframe as sufficient.... As the objective of the forecast is predicting CO_2 concentration for the next hour, all evaluated algorithms are trained to forecast the next 12 values. Because explainability is the main focus of this study, we evaluated accuracy to compare the predictive power of different algorithms without the aim of optimizing it, . The results of our evaluation of forecasting algorithms on the CO_2 concentration data are shown in Table 2. We compare the performance of the algorithms based on the metrics *mean absolute error* (MAE) and *root mean squared error* (RMSE), as defined in [13], where each metric represents an error, with lower values indicating higher performance. N-BEATS shows the lowest error on both metrics, showing that the algorithm performs best in our CO_2 concentration forecasting scenario. Non-Stationary Transformer and D-Linear show the second-highest performance, while ARIMA and DEPTS have the highest errors.

From the evaluation it has become clear that N-BEATS is the most suitable AI algorithm for the forecasting component in our digital twin of a smart building.

Table 2: Evaluation of AI algorithm performance for the CO_2 concentration forecasting task. Best performing algorithm by metric highlighted in bold.

Algorithm	MAE	RMSE
N-BEATS	**30.52**	**66.3**
D-Linear	31.64	70.37
NST	31.11	68.79
ARIMA	35.53	92.59
DEPTS	44.54	84.39

Fig. 2 shows a sample of the dashboard d, showing the input stream of CO_2 concentration data i_C as well as the underlying ground truth for the next hour. The figure does not represent the complete dashboard, as it does not display the number of outliers k and the model explanations e_i or e_p. In addition to the input data, the forecasts made by the five evaluated algorithms are shown. As the ground truth would not be available in a production setting with live data, in this case, the dashboard would only show historical data and the forecast of the employed forecasting algorithm. When using a non-explainable algorithm, the forecast trajectory of CO_2 (colored lines) is the only output of the algorithm, as shown in the figure, while no further details are provided on "why" that particular forecast has been made. On the contrary an explainable algorithm will provide additional explanations on why the forecasted values follow a particular trajectory. Such explanations can be to link the forecasted values to historical values, patterns, or trends (e.g., upward-going or seasonal) that have occurred in the past and are repeating in the future, such as the ones depicted in Figure 3.

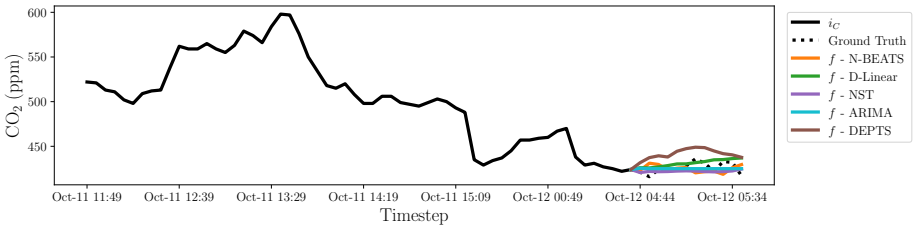


Fig. 2: Sample of the dashboard d showing the input stream i_C, and forecast f made by different machine learning models.

3.4 Explainability of Methods for Operator's Dashboard

In this section, we investigate the explainability of the forecasting methods, which can be either inherent to the AI algorithm or applied post hoc within the DT model. We follow the definitions of Molnar [23] regarding explainability properties, characterizing them as applicable for each method. As the not inherently explainable methods solely provide the numeric values of their output without a rationale behind it, we apply TS-MULE [30], a post hoc explainability method extending LIME [28] for time series data. The post hoc explainability method represents the explainability component c_x for opaque algorithms.

Fig. 3 showcases the decomposition of the forecast by N-BEATS into trend and seasonality elements. Trend provides a general direction of the data, showing a long-term increase or decrease in value, while seasonality represents cyclic patterns such as week-weekend cycles. The model's forecast can be obtained by adding the two elements of trend and seasonality. The visualization is based on the decomposition of the prediction as seen in [24]. The forecast is based on a

sample from the smart building CO_2 concentration data we use for evaluation in this section. Notably, both elements have a different scale, with the trend having higher absolute values than the seasonality, while the seasonality shows more change over time. This offers a higher degree of comprehensibility for a human operator, different from a non-explainable forecast, where the prediction is solely based on a nonlinear combination of neural network weights and intermediate features. However, comprehensibility is hard to measure, and a qualitative analysis based on human judgment would be necessary to make a conclusive statement on this property. The explanation of N-BEATS is local, representing an individual prediction while not showing the model's certainty or indicating novelty. The fidelity of the explanation is high, as it represents a decomposition, which is equivalent to the actual forecast of the model. However, as the explanation can not be used to predict unseen data, this type of explanation offers low accuracy. During our experiments, it became clear that the stability of N-BEATS is high, as its explanations vary little when perturbing the input.

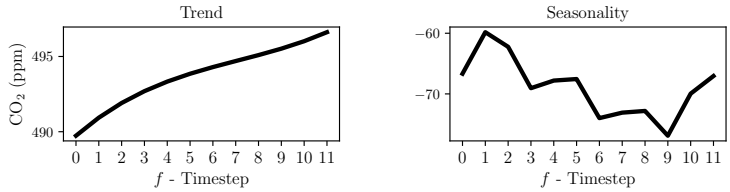


Fig. 3: Inherent explanations e_i provided by N-BEATS, decomposing its forecast into trend and seasonality components.

Another kind of visualization for inherent explanations of the deep learning algorithm DEPTS [7] is shown in Fig. 4, following the approach presented in the original paper. Similar to N-BEATS, DEPTS provides prediction-based explanations based on a periodic element (DEPTS-P) and a local element (DEPTS-L), leading to a low degree of representativeness. Different from the visualization of N-BEATS, the two elements are merged in this graph, sharing the same scale, leading to a different visualization style. DEPTS shows a stable periodicity, indicating that the data do not experience significant cyclic patterns. The local component of DEPTS shows lower absolute values than the periodic component. Overall, the decomposition-based explanation approaches of N-BEATS and DEPTS are similar, basing their forecast on two elements that sum up to the algorithms' forecast. It can be argued that the visualization style of DEPTS' explanations provides a lower degree of comprehensibility than N-BEATS, as both positive and negative values are merged in one figure. Further, similar to N-BEATS, the explanation does not provide certainty or novelty measures, while keeping a high level of fidelity due to the decomposition-based approach. The stability of DEPTS' explanations is high, but it does not present feature importance for the input.

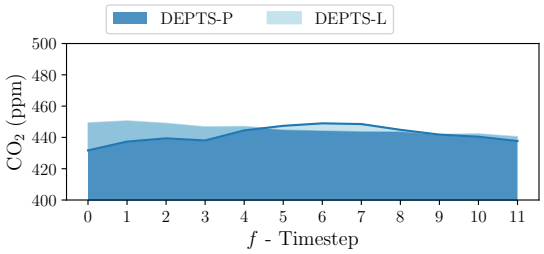
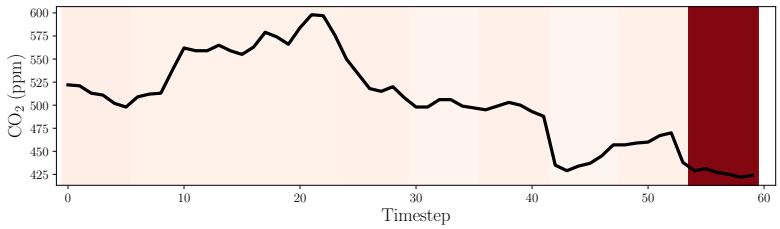


Fig. 4: Inherent explanations e_i provided by DEPTS, decomposing its forecast into periodic (DEPTS-P) and local (DEPTS-L) elements.

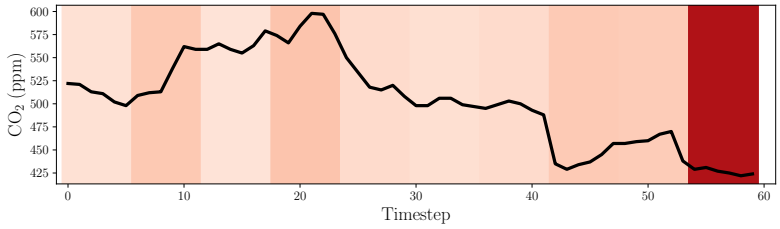
We apply the post hoc explainability method TS-MULE [30], which is a generalization of LIME for time series, to the three best-performing forecasting methods D-Linear, N-BEATS, and Non-Stationary Transformer, as seen in Fig. 5. This represents the post hoc explainability component c_x, which generates post hoc explanations e_p based on the forecast and the model used in c_f. TS-MULE uniformly segments the input data and assigns relevance scores to each segment based on its influence on the prediction of the forecasting algorithm. In the figure, darker segments indicate higher relevance for the respective algorithm, while lighter segments indicate lower relevance. By integrating the post hoc explanations with the dashboard, it acts as an interface for explainability, providing explanations for the human operator of the digital twin.

Different from the inherent explanations of N-BEATS and DEPTS, TS-MULE works on an input level, which provides a higher degree of comprehensibility for the operator, as they can judge which past timesteps were critical for the forecasting model, giving a degree of importance for each segment. However, as the relevance is assigned based on input perturbation, the fidelity of the explanations is lower. As the explanations do not provide confidence intervals or other uncertainty measures, TS-MULE does not provide a degree of certainty or novelty regarding the predictions. Similar to N-BEATS and DEPTS, TS-MULE has a low degree of representativeness, as it provides explanations for an individual prediction and does not characterize the forecasting model as a whole. In our experiments, TS-MULE was averaged over 100 runs, as the method has low stability and can provide differing results for the same input.

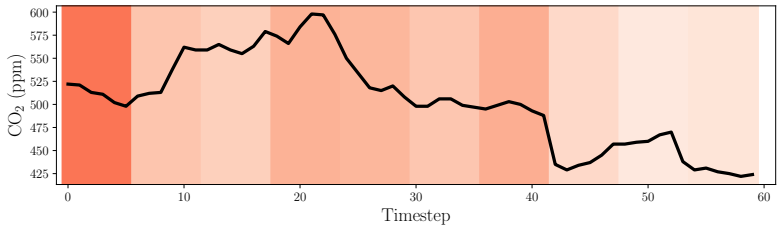
Both D-Linear (5a) and N-BEATS (5b) have the highest relevance score in the last segment of the input data, showing that the algorithms base their forecasts on the most recent observations. Non-Stationary Transformer (5c) shows the opposite behavior, as it reaches the highest relevance score on the first segment, gradually giving less relevance to further segments, while the last two segments receive the lowest overall relevance. In a production setting, the explanations shown in Fig. 5 are integrated with the dashboard d, which shows both the forecast of the AI algorithm and the explanations, which are applied post hoc by using TS-MULE.



(a) Post hoc explanations e_p of D-Linear with TS-MULE



(b) Post hoc explanations e_p of N-BEATS with TS-MULE



(c) Post hoc explanations e_p of Non-Stationary Transformer with TS-MULE

Fig. 5: Sample of post hoc explanations e_p of the explainability component c_x. The algorithm TS-MULE [30] is used for segment-based relevance, where darker segments indicate higher relevance of the data on the algorithm's prediction.

4 Discussion

Usefulness of explanations - In the previous section, we introduced an example of a digital twin of a smart building, working solely on the basis of CO_2 concentration data from one type of sensor. The presented explainability methods, both inherent and post hoc, show in different ways how the respective AI algorithms arrived at their projected forecasts. Whether these explanations are useful to an operator depends on the domain and the use case and needs to be evaluated on a case-by-case basis. Table 3 summarizes the explainability properties of the evaluated explainability methods. The usefulness of an explanation is highly dependent on the *comprehensibility* of the explanation, as an incomprehensible explanation does not benefit the operator of the digital twin. As comprehensibility is subjective, a degree of comprehensibility can be estimated with qualitative approaches involving human feedback. Further, the *fidelity* of

an explanation is key for the operator, as every explanation needs to accurately reflect the underlying behavior of the model. The inherent explanations of N-BEATS and DEPTS provided a higher level of fidelity than TS-MULE, as they are specific to the model's internal processing rather than calculated post-hoc based on perturbations. We further characterized the properties of *representativeness* and *certainty*, and *novelty*, which were not fulfilled by both the inherent explainability methods of N-BEATS and DEPTS and the post-hoc method TS-MULE. However, it can be argued that these properties are not necessary for explanations and only provide additional value when present. During our evaluation, We first benchmarked the accuracy of multiple AI algorithms, as that is the primary quality measure important for a digital twin. In the scenario where multiple algorithms yield comparable accuracy, inherent explainability is key, making white-box algorithms with explainability properties preferable. Additionally, we evaluated the *stability* of each explanation method during our experiments, concluding that TS-MULE has a lower level of stability than the other methods.

Table 3: Explainability properties of the evaluated explainability methods.*

Explainability Method	Inherent	Fidelity	Stability	Comprehensibility
N-BEATS	Inherent	High	High	Subjective
DEPTS	Inherent	High	High	Subjective
TS-MULE	Post-hoc	Low	Low	Subjective

Explainability Method	Certainty	Novelty	Degree of importance	Representativeness
N-BEATS	Low	Low	Low	Low
DEPTS	Low	Low	Low	Low
TS-MULE	Low	Low	High	Low

* Accuracy and consistency were excluded from the table, as they are not applicable in our case

Architectural Model - The focus of this work is on the definitions and the model of the proposed digital twin for establishing a formal and conceptual basis for integrating explainable AI. Due to this, our evaluation covered the AI component, comparing the accuracy of different algorithms as the essential DT aspect, as well as focusing to the explainability aspect, for investigating and integrating both inherent and post hoc explanations. The introduced rule-based component, representing the feedback loop to the physical system, was not investigated for explainability in this study as it does not work on the basis of AI and therefore also does not have a connection to this quality aspect.

Real-world setting complexity - In the study, we limited the use case to the analysis of a single variable (CO_2), to emphasise the core contribution - integration of XAI to the digital twin model. In a real-world setting, a digital twin of a smart building should include additional variables such as temperature,

humidity, occupancy, or light intensity. With access to more data, AI algorithms perform better, allowing them to make more accurate forecasts, thereby improving the overall performance of the digital twin. As the complexity of the system increases, the complexity of explanations also increases, requiring the explainability methods to scale with the digital twin. In addition, the digital twin would have more control over the physical system, for example, by adapting the heating system based on occupancy and projected changes in temperature. In this case, an explanation could cover both trends in past temperature and occupancy, providing a more detailed summary of the underlying patterns for the operator.

Generalizability - The architectural model of D_s presented in section 3.2, even specific to our presented use case on CO_2 concentration forecasting in smart buildings, it has been meant for avoiding further details to put forward generalisation. Our proposed component-based notation can be generalized to digital twins of any domain, a similar model can be created for any digital twin D. An architectural model helps visualize the internal flow of information within the DT while also delineating the capabilities of the different components. Still, it is our intention to, in the future study, detail the presented core architecture components by the means of a meta-model.

5 Conclusion and Future Work

In this study, we introduced the concept of explainable AI in digital twins, outlining a use case in smart buildings where an AI algorithm is used to forecast CO_2 concentration. To illustrate our digital twin, we presented an architectural model of the system, showing its interaction with the smart building and a human operator. For the proposed digital twin, we evaluated five AI algorithms, comparing their accuracy in forecasting CO_2 concentration. The deep learning algorithm N-BEATS showed the highest performance in forecasting, indicating that it is the most suitable candidate for our digital twin. We further investigated the explainability of the evaluated AI algorithms, outlining both inherently provided model explanations and post hoc explanations based on TS-MULE.

With the definitions given in this paper, we are planning to further investigate the outlined use case, making use of more sensors and connecting the digital twin to the physical system in real-time. In this more complex scenario, explainability methods must be adapted to the available data showing correlations between variables. To evaluate the practicality of the provided explanations for the DT, we are planning to conduct a qualitative analysis based on operator feedback.

In future work, we are planning to empirically investigate machine learning explainability in digital twins based on a user study. As explainability properties are generally assessed qualitatively and some, such as comprehensibility, can be subjective, a user study could contribute to this line of research.

Acknowledgements

We would like to thank Atrium Ljungberg AB for providing the data for the evaluation conducted in this study.

References

1. Apostolidis, A., Stamoulis, K.P.: An ai-based digital twin case study in the mro sector. Transportation Research Procedia **56**, 55–62 (2021)
2. Beaudouin, V., Bloch, I., Bounie, D., Clémençon, S., d'Alché Buc, F., Eagan, J., Maxwell, W., Mozharovskyi, P., Parekh, J.: Flexible and context-specific ai explainability: a multidisciplinary approach. arXiv preprint arXiv:2003.07703 (2020)
3. Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M.: Time series analysis: forecasting and control. John Wiley & Sons (2015)
4. Brenner, B., Hummel, V.: Digital twin as enabler for an innovative digital shopfloor management system in the esb logistics learning factory at reutlingen-university. Procedia Manufacturing **9**, 198–205 (2017)
5. Burkart, N., Huber, M.F.: A survey on the explainability of supervised machine learning. Journal of Artificial Intelligence Research **70**, 245–317 (2021)
6. Clausen, A., Arendt, K., Johansen, A., Sangogboye, F.C., Kjærgaard, M.B., Veje, C.T., Jørgensen, B.N.: A digital twin framework for improving energy efficiency and occupant comfort in public and commercial buildings. Energy Informatics **4**, 1–19 (2021)
7. Fan, W., Zheng, S., Yi, X., Cao, W., Fu, Y., Bian, J., Liu, T.Y.: Depts: deep expansion learning for periodic time series forecasting. arXiv preprint arXiv:2203.07681 (2022)
8. Fukas, P., Rebstadt, J., Menzel, L., Thomas, O.: Towards explainable artificial intelligence in financial fraud detection: Using shapley additive explanations to explore feature importance. In: International Conference on Advanced Information Systems Engineering. pp. 109–126. Springer (2022)
9. Gallala, A., Kumar, A.A., Hichri, B., Plapper, P.: Digital twin for human–robot interactions by means of industry 4.0 enabling technologies. Sensors **22**(13), 4950 (2022). https://doi.org/https://doi.org/10.3390/s22134950
10. Gruver, N., Finzi, M., Qiu, S., Wilson, A.G.: Large language models are zero-shot time series forecasters. arXiv preprint arXiv:2310.07820 (2023)
11. Guidotti, R.: Counterfactual explanations and how to find them: literature review and benchmarking. Data Mining and Knowledge Discovery pp. 1–55 (2022)
12. Huang, Z., Shen, Y., Li, J., Fey, M., Brecher, C.: A survey on ai-driven digital twins in industry 4.0: Smart manufacturing and advanced robotics. Sensors **21**(19), 6340 (2021)
13. Hyndman, R.J., Koehler, A.B.: Another look at measures of forecast accuracy. International journal of forecasting **22**(4), 679–688 (2006)
14. Kapteyn, M.G., Knezevic, D.J., Willcox, K.: Toward predictive digital twins via component-based reduced-order models and interpretable machine learning. In: AIAA Scitech 2020 Forum. p. 0418 (2020)
15. Kapteyn, M.G., Willcox, K.E.: From physics-based models to predictive digital twins via interpretable machine learning. arXiv preprint arXiv:2004.11356 (2020)
16. Kobayashi, K., Alam, S.B.: Explainable, interpretable, and trustworthy ai for an intelligent digital twin: A case study on remaining useful life. Engineering Applications of Artificial Intelligence **129**, 107620 (2024)
17. Kreuzer, T., Papapetrou, P., Zdravkovic, J.: Artificial intelligence in digital twins—a systematic literature review. Data & Knowledge Engineering **151**, 102304 (2024). https://doi.org/https://doi.org/10.1016/j.datak.2024.102304, https://www.sciencedirect.com/science/article/pii/S0169023X24000284

18. Li, B., Liu, Y., Tan, L., Pan, H., Zhang, Y.: Digital twin assisted task offloading for aerial edge computing and networks. IEEE Transactions on Vehicular Technology **71**(10), 10863–10877 (2022)
19. Liu, S., Yu, H., Liao, C., Li, J., Lin, W., Liu, A.X., Dustdar, S.: Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In: International conference on learning representations (2021)
20. Liu, Y., Wu, H., Wang, J., Long, M.: Non-stationary transformers: Exploring the stationarity in time series forecasting. Advances in Neural Information Processing Systems **35**, 9881–9893 (2022)
21. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. Advances in neural information processing systems **30** (2017)
22. Matulis, M., Harvey, C.: A robot arm digital twin utilising reinforcement learning. Computers & Graphics **95**, 106–114 (2021)
23. Molnar, C.: Interpretable machine learning. Lulu. com (2020)
24. Oreshkin, B.N., Carpov, D., Chapados, N., Bengio, Y.: N-beats: Neural basis expansion analysis for interpretable time series forecasting. arXiv preprint arXiv:1905.10437 (2019)
25. Patel, S.S.: Explainable machine learning models to analyse maternal health. Data & Knowledge Engineering p. 102198 (2023)
26. Qi, Q., Tao, F., Hu, T., Anwer, N., Liu, A., Wei, Y., Wang, L., Nee, A.: Enabling technologies and tools for digital twin. Journal of Manufacturing Systems **58**, 3–21 (2021)
27. Rathore, M.M., Shah, S.A., Shukla, D., Bentafat, E., Bakiras, S.: The role of ai, machine learning, and big data in digital twinning: A systematic literature review, challenges, and opportunities. IEEE Access **9**, 32030–32052 (2021). https://doi.org/10.1109/ACCESS.2021.3060863
28. Ribeiro, M.T., Singh, S., Guestrin, C.: " why should i trust you?" explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1135–1144 (2016)
29. Ribeiro, M.T., Singh, S., Guestrin, C.: Model-agnostic interpretability of machine learning. arXiv preprint arXiv:1606.05386 (2016)
30. Schlegel, U., Vo, D.L., Keim, D.A., Seebacher, D.: Ts-mule: Local interpretable model-agnostic explanations for time series forecast models. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 5–14. Springer (2021)
31. Singh, M., Fuenmayor, E., Hinchy, E.P., Qiao, Y., Murray, N., Devine, D.: Digital twin: Origin to future. Applied System Innovation **4**(2), 36 (2021)
32. Suhail, S., Iqbal, M., Hussain, R., Jurdak, R.: Enigma: An explainable digital twin security solution for cyber–physical systems. Computers in Industry **151**, 103961 (2023)
33. Tao, F., Xiao, B., Qi, Q., Cheng, J., Ji, P.: Digital twin modeling. Journal of Manufacturing Systems **64**, 372–389 (2022)
34. Tao, F., Zhang, H., Liu, A., Nee, A.Y.: Digital twin in industry: State-of-the-art. IEEE Transactions on industrial informatics **15**(4), 2405–2415 (2018)
35. Wang, Z., Gupta, R., Han, K., Wang, H., Ganlath, A., Ammar, N., Tiwari, P.: Mobility digital twin: Concept, architecture, case study, and future challenges. IEEE Internet of Things Journal **9**(18), 17452–17467 (2022)
36. Xie, X., Parlikad, A.K., Puri, R.S.: A neural ordinary differential equations based approach for demand forecasting within power grid digital twins. In: 2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm). pp. 1–6. IEEE (2019)

37. Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are transformers effective for time series forecasting? In: Proceedings of the AAAI conference on artificial intelligence. vol. 37, pp. 11121–11128 (2023)
38. Zhou, G., Zhang, C., Li, Z., Ding, K., Wang, C.: Knowledge-driven digital twin manufacturing cell towards intelligent manufacturing. International Journal of Production Research **58**(4), 1034–1051 (2020)
39. Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R.: Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In: International conference on machine learning. pp. 27268–27286. PMLR (2022)