Published: 22 September 2024

Citation in BibTeX format

MODELS Companion '24: ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems
September 22 - 27, 2024
Linz, Austria

Conference Sponsors:
SIGSOFT

RESEARCH-ARTICLE

A Digital Twin System for Oil And Gas Industry: A Use Case on Mooring Lines Integrity Monitoring

VINICIUS KREISCHER DE ALMEIDA

DOUGLAS ERICSON DE OLIVEIRA

CLAUDIO DANIEL DE BARROS

GABRIEL DOS SCATENA, University of São Paulo, Sao Paulo, SP, Brazil

ASDRUBAL N QUEIROZ FILHO, University of São Paulo, Sao Paulo, SP, Brazil

FÁBIO LEVY SIQUEIRA, University of São Paulo, Sao Paulo, SP, Brazil

View all

Open Access Support provided by:

Pontifical Catholic University of Rio de Janeiro

University of São Paulo

Petrobras

Federal Center for Technological Education Celso Suckow da Fonseca

A Digital Twin System for Oil And Gas Industry: A Use Case on Mooring Lines Integrity Monitoring

Vinicius Kreischer de Almeida
National Laboratory of Scientific Computing
Petropolis, Brazil
valmeida@lncc.br

Douglas Ericson de Oliveira
National Laboratory of Scientific Computing
Petropolis, Brazil
ericson@lncc.br

Claudio Daniel T. de Barros
Petrobras and National Laboratory of Scientific Computing
Rio de Janeiro, Brazil
claudiodtbarros@gmail.com

Gabriel dos Santos Scatena
Universidade de São Paulo
São Paulo, Brazil
gabrielscatena@usp.br

Asdrubal N. Queiroz Filho
Universidade de São Paulo
São Paulo, Brazil
asdrubal@usp.br

Fábio Levy Siqueira
Universidade de São Paulo
São Paulo, Brazil
levy.siqueira@usp.br

Anna Helena Reali Costa
Universidade de São Paulo
São Paulo, Brazil
anna.reali@usp.br

Edson Satoshi Gomi
Universidade de São Paulo
São Paulo, Brazil
gomi@usp.br

Leonardo A. F. Mendoza
PUC-Rio
Rio de Janeiro, Brazil
mendonza@ele.puc-rio.br

Evelyn C. S. Batista
PUC-Rio
Rio de Janeiro, Brazil
evelyn@puc-rio.br

Cristian E. Muñoz
PUC-Rio
Rio de Janeiro, Brazil
crismunoz@puc-rio.br

Isabela G. Siqueira
Petrobras
Rio de Janeiro, Brazil
isabela.siqueira@petrobras.com.br

Rodrigo A. Barreira
Petrobras
Rio de Janeiro, Brazil
barreira@petrobras.com.br

Ismael H. F. dos Santos
Petrobras
Rio de Janeiro, Brazil
ismaelh@petrobras.com.br

Carlos Cardoso
National Laboratory of Scientific Computing
Rio de Janeiro, Brazil
cardoso@lncc.br

Eduardo Ogasawara
CEFET/RJ
Rio de Janeiro, Brazil
eogasawara@ieee.org

Fabio Porto
National Laboratory of Scientific Computing
Petropolis, Brazil
fporto@lncc.br

ABSTRACT

A Digital Twin is a virtual representation of a real-world object or process, leveraging powerful computational architectures available both on-premises and in the cloud. By harnessing the increased availability of real-time data and advancements in machine learning predictive algorithms, Digital Twins find applications across various domains such as Earth Science, Oil and Gas, and Healthcare. However, realizing their full potential demands addressing the technical complexities of integrating numerous components during development and operational phases of the system. This paper describes an ongoing effort to build a comprehensive platform that supports the entire lifecycle of a Digital Twin, from continuous specialized model training to online prediction and event detection, by capturing and processing live data. This approach enables timely updates to the virtual representations of physical elements within the twin application as they change. We detail each component of the Digital Twin solution and demonstrate its applicability through a real use case implemented in the Oil and Gas industry. Specifically, we focus on monitoring the motion of oil platforms to ensure the integrity of the mooring systems and respond to adverse conditions through an alert system powered by our platform.

CCS CONCEPTS

• **Computer systems organization** → *Real-time system architecture*; • **Information systems** → **Decision support systems**; •

MODELS Companion '24, September 22–27, 2024, Linz, Austria

ACM ISBN 979-8-4007-0622-6/24/09
https://doi.org/10.1145/3652620.3688244

Computing methodologies → *Machine learning*; **Simulation support systems**.

KEYWORDS

Digital Twin, Twinscie, Model Management, Data Management, Stream Data, Mooring Lines, Integrity, Monitoring, Oil and Gas, Offshore Platforms

ACM Reference Format:

Vinicius Kreischer de Almeida, Douglas Ericson de Oliveira, Claudio Daniel T. de Barros, Gabriel dos Santos Scatena, Asdrubal N. Queiroz Filho, Fábio Levy Siqueira, Anna Helena Reali Costa, Edson Satoshi Gomi, Leonardo A. F. Mendoza, Evelyn C. S. Batista, Cristian E. Muñoz, Isabela G. Siqueira, Rodrigo A. Barreira, Ismael H. F. dos Santos, Carlos Cardoso, Eduardo Ogasawara, and Fabio Porto. 2024. A Digital Twin System for Oil And Gas Industry: A Use Case on Mooring Lines Integrity Monitoring. In *ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems (MODELS Companion '24), September 22–27, 2024, Linz, Austria.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3652620.3688244

1 INTRODUCTION

Digital Twins (DT) represent a significant breakthrough, enabling the creation of virtual representations of physical objects or processes [21, 39]. These real-time digital counterparts provide predictive insights, enable proactive maintenance, and optimize operational performance through continuous monitoring and data analysis [12]. Their relevance and adoption is rapidly growing across multiple domains, such as Healthcare, Robotics, Smart Cities, and Energy. By integrating the Internet of Things (IoT), cloud computing, and Artificial intelligence (AI), DTs offer transformative potential for managing and optimizing complex systems and processes across various industries.

In the oil and gas industry, DT applications are particularly suitable for monitoring Floating Production Storage and Offloading (FPSO) units [11]. These units are crucial to offshore operations, and their performance, security, and longevity depend on the integrity of their mooring lines, among other factors. In this context, DTs provide actionable insights by continuously collecting, processing, and integrating data from sensors and IoT devices [10]. This enables early detection of issues, optimized maintenance schedules, and failure predictions, thereby improving decision-making and reducing operational costs and risks.

Despite its potential, implementing and operating a DT platform for monitoring FPSO mooring lines presents several significant challenges. The primary issue involves developing and integrating the multiple components necessary to meet the application's requirements. Firstly, the system must efficiently handle real-time data ingestion, processing, and integration while ensuring this data is easily accessible to subsequent components. Secondly, leveraging reliable machine learning models to provide predictive insights on selected metrics is crucial. However, managing the lifecycle of these machine learning models is complex, requiring careful handling of related artifacts such as datasets, algorithms, and predictions [33]. Furthermore, the system must accommodate different computation environments to meet the diverse computational requirements of various machine learning lifecycle steps. This includes ensuring that the environments can support data preprocessing, model training, and real-time inference, each with specific demands. Addressing these challenges is essential for realizing the full benefits of a Digital Twin platform in enhancing the safety and efficiency of FPSO operations.

To address these challenges, we propose a comprehensive Digital Twin platform designed to support the entire lifecycle of FPSO mooring line integrity monitoring. This platform comprises various key components, each developed to tackle the diverse challenges previously discussed. The first of these components is a Stream Data Manager, responsible for capturing, processing, and integrating real-time data from sensors on the FPSOs. This ensures the platform can efficiently manage the high volume and velocity of data generated by these sensors, providing a robust foundation for real-time monitoring. Another crucial data-related component of the platform is a Domain Data Manager, which manages efficient access to non-real-time data, encompassing the characteristics and contextual information of domain objects.

The third key component is a machine learning management system called Twinscie, which manages the lifecycle of machine learning models and related artifacts, such as datasets and algorithms. Twinscie also interfaces with the various computational environments used for different stages of the machine learning process, ensuring that the platform can meet the specific computational requirements of data processing, model training, and real-time inference. Moreover, Twinscie provides a dataflow language that allows users to define machine learning tasks as a sequence of independent activities. By means of Twinscie, the proposed DT platform leverages machine learning models developed specifically for monitoring FPSO mooring lines.

Finally, the Twin Application component manages the digital representation of the real twinned application. It composes the current status of the digital reflection using measured observations captured by sensors and events detected by the pre-trained ML models managed by Twinscie component. By integrating this real-time data with the domain-specific characteristics and contextual information, the Twin Application ensures a comprehensive and accurate digital reflection of the physical system. This detailed and dynamic representation allows for continuous monitoring and analysis, ultimately enabling proactive maintenance strategies.

As part of our ongoing efforts, key functionalities of the proposed DT platform have been developed and tested at Petrobras, a leading Brazilian oil and gas company, within their on-premises environment to support the monitoring of FPSO mooring lines. The initial results are promising in terms of operational efficiency and suggest the potential for further enhancements, ensuring the robustness and adaptability of the platform to a wider range of scenarios and requirements in mooring line integrity monitoring.

The remainder of this work is structured as follows: Section 2 reviews related work. Section 3 provides an overview of the proposed Digital Twin computational platform's architecture and components. The Twinscie component is detailed in Section 4. Section 5 describes how stream and domain data are handled within the platform. Section 6 introduces developed machine learning models for FPSO mooring system, while Section 7 demonstrates the integration of one of these models with Twinscie and presents related results. Finally, Section 8 offers conclusions and discusses perspectives for future work.

2 RELATED WORK

Digital Twin (DT) systems have rapidly progressed from a theoretical concept envisioned by early research [37] to powerful tools impacting various fields. Advancements in AI, IoT, and cloud computing have enabled the development of sophisticated DTs capable of real-time monitoring, analysis, and optimization [9, 23]. These technological advancements have led to the successful implementation of DTs in diverse sectors, ranging from energy [18] to healthcare, where they are used for personalized medicine, remote patient monitoring with wearable devices, and efficient clinical trials [17, 38]. Furthermore, DTs are transforming industrial robotics by enabling real-time optimization of control systems for enhanced precision and adaptability [22]. In smart cities, DTs are leveraged for traffic management, optimizing resource allocation, and enhancing public safety [16].

Recent studies highlight the adoption of DTs in the oil and gas industry for optimizing operations, minimizing risks, and improving asset management while enhancing productivity and safety in project lifecycles [41]. DTs are being integrated into diverse offshore oil and gas production aspects, including drilling, production processes, equipment maintenance, and oilfield asset management [1]. Successfully implementing these complex DT systems in real-world settings requires robust cloud and edge computing infrastructure capable of handling the substantial computational and data demands [20].

DTs are proving particularly valuable in optimizing drilling operations, enabling real-time monitoring of drilling parameters, automated adjustments to drilling plans, and early detection of potential problems like stuck pipes or wellbore instability. Mayani et al. [2] present a drilling well DT that integrates real-time data and advanced modeling to optimize operations, reduce downtime, and enhance safety. Similarly, Thomas and Ziatdinov [5] advocate for a DT-driven methodology to improve drilling processes, emphasizing performance enhancement, non-productive time reduction, and maximizing oil and gas production.

DTs are also effective in optimizing production processes and enhancing oilfield asset management. Shen et al. [35] show that integrating real-time data from IoT sensors with AI algorithms within a DT framework can optimize production parameters, predict equipment failures, and enable data-driven decision-making for enhanced production efficiency. Furthermore, Lai et al. [4] demonstrate how combining DTs with big data analytics can provide a holistic view of oilfield operations, enabling operators to identify production bottlenecks, optimize resource allocation, and proactively manage asset integrity for improved performance and profitability.

Furthermore, DTs enable predictive maintenance of critical equipment, such as pumps, compressors, and turbines, by leveraging simulation data and machine learning techniques to predict the Remaining Useful Life (RUL) of components [6]. By forecasting potential equipment failures, operators can optimize maintenance schedules, procure spare parts in advance, and minimize costly unplanned downtime, ultimately enhancing equipment reliability and reducing operational risks.

Concurrently, visual inspection techniques are integrated into DTs, leveraging image processing and deep learning algorithms to automate defect detection from marine images [19]. This integration enables the identification of defects such as corrosion or cracks, significantly improving the efficiency and accuracy of inspections and providing valuable data for the DT model.

Mooring system integrity is critical for the safe operation of offshore structures like FPSOs, and the rise of sensor data and machine learning has fueled the development of DTs for comprehensive mooring system management. These DTs leverage real-time sensor data, historical information, physics-based models, and advanced machine learning algorithms to create virtual representations of physical mooring systems, enabling continuous monitoring and assessment of their health [7, 30]. Sa'ad et al. [8] employed a neural network and classifier combination to predict FPSO motion and assess mooring line failure probability.

Implementing DTs in the oil and gas industry has shown great potential in enhancing the monitoring and management of mooring systems, which are critical for the stability and performance of offshore structures [29]. DTs provide a sophisticated means of combining real-time data with virtual models to predict and analyze the behavior of these systems under various conditions. Despite the promise of DTs, challenges remain, particularly in their ability to generalize across diverse environmental conditions and manage the vast amounts of data involved in real-time monitoring.

Recent research efforts have focused on overcoming these limitations through domain generalization and transfer learning techniques and integrating advanced visual inspection methods using image processing and deep learning algorithms [3]. This integration improves the accuracy and efficiency of defect detection and enhances the overall robustness of DT-based monitoring systems.

While promising, these data-driven approaches often face challenges in generalizing to new environmental conditions. Ongoing efforts are exploring domain generalization and transfer learning techniques to overcome these limitations and enhance the robustness of DT-based mooring system monitoring [43]. Alternatively, Ribeiro et al. [31] discuss constructing subset models based on learned data partitions. The latter approach finds similarities among different platform behaviors and ocean conditions, building subset models on data partition that reflect common behavior. The subset approach leads to multiple candidate models whose application for a given inference scenario leverages the knowledge about the training data distribution, as discussed in [26].

This work builds upon these advancements in DT technology by proposing a novel system specifically designed for the oil and gas industry, focusing on mooring line integrity monitoring as an important use case. Our system addresses challenges related to computational complexity, data integration, and capturing the complex interactions between mooring lines, platform dynamics, and environmental factors. By utilizing a multi-model approach that integrates specialized machine learning models for specific failure modes and anomalies, coupled with a robust data management infrastructure and hybrid modeling strategy, our system provides comprehensive and accurate real-time monitoring for enhanced mooring system integrity management.

3 ARCHITECTURE OVERVIEW

Figure 1 depicts the proposed digital twin system architecture. At the top of the figure stands the Twin Application component, where the digital representation of the real twinned application is managed. The information required to compose the current status of the digital reflection includes the measured observations captured by sensors, characteristics and contextual information of domain objects, and the events detected by the pre-trained ML models (see Section 6). The data are obtained by integration interfaces between the Twin Application and the Twinscie system. The latter receives the incoming observations from the Stream Data Management (SDM) system, as well as domain data from the Domain Data Manager (DDM) runs pre-processing dataflows to validate and structure the data according to the ML models input data formats, and invokes the models. The models can produce a list of detected anomalous events or predictions over the input data. Once this process has been executed, the measured observations and the models' results feed the Twin Application component. Finally, the SDM system accesses the sensor's control devices to collect real-time data, updating the status of the mirrored application. The SDM system collects the data streams (see Section 5), pre-processes, and materializes the resulting data into a database system. Twinscie periodically queries the SDM system for new data to refresh the Twin Application state.

4 THE TWINSCIE SYSTEM

Twinscie is a machine learning management system supporting the entire ML life cycle [14, 27, 34]. As a component of the digital twin system, Twinscie provides access to managed artifacts through the invocation of services made available through its RESTful application program interface (API) and web interface. The API is important for establishing a communication protocol that other DT components use to request services programmatically. Conversely, the web interface allows system administrators to interact with the system's services. Additionally, Twinscie provides a dataflow language, which can represent data transformation, training, and prediction processes as a composition of discrete activities.

Figure 2 depicts the general architecture of the Twinscie system. The system maintains a catalog for artifacts metadata management, including provenance information. Twinscie also includes the concept of available *Environments*, which refers to an infrastructure where tasks can be scheduled to run. An *Environment* includes a computational system (e.g., Big data cluster, AI Workstation, supercomputer, etc.), storage for managed artifacts, and a *MLFlow* [44] instance that isolates Twinscie from the heterogeneity found in different ML engines supported by the system: TensorFlow, pytorch, scikit-learn, Keras, etc.

The main services that the Twinscie system provides are:

Register a Dataset: all datasets known to the system must be registered. The action leads Twinscie to save the necessary information to access the corresponding dataset in the catalog. Observe that a dataset may have been produced elsewhere and need to be registered, or it could have been produced by a Twinscie dataflow, in which case the system automatically registers it.

Register a Learner: In the context of the Twinscie system, learners are a package of Python scripts, library dependencies, and execution information used to train a model.

Register a Model: The registration of a model is achieved by either training or importing actions. The training of a model is performed by informing a particular learner, a registered dataset, and a set of hyperparameters' values.

Set and Run a Dataflow: Twinscie supports the specification of an abstract dataflow that describes the graph of operation dependencies. In preparation for running, an abstract dataflow is associated with input datasets registered in Twinscie. It is scheduled to run in one of the available Twinscie Environments.

Run an Inference: Once a model has been registered, it can be invoked with an input dataset to compute the predictions. Twinscie supports different tasks, including event detection, which is interesting for the *Twin Application.*

Inform the twin application: The system manages the data obtained from sensors and the predictions produced by models. Additionally, metadata information regarding the processes, including statistics and metrics, are available for consumption by the Twin Application through the system REST API.

4.1 Twinscie Users

Twinscie is designed to support users in various roles, each with distinct responsibilities concerning its services. The *Administrator* manages the system objects, manually registers artifacts and metatada information, and specifies environments. The *Model Developer* registers *learners* and uses them to build the models. This user often writes the code for the *Learner* and creates *dataflows* for data preparation and model training. The *Application Developer* integrates Twinscie with its application by calling the appropriate Twinscie API.

4.2 Model Management

Twinscie enables a model to be constructed using other frameworks and have it imported into the system. Alternatively, the *Model developer* can use the provided set of services to collect the necessary data, process it, and feed it to construct a model, which is performed through the training of a *Learner* previously registered to the system. Using the same *Learner* with a different dataset and hyperparameters' values leads to a different model version. The system tracks the relationship between previous versions and a recently created one. Once a model has been built, it can compute inferences. In a DT, an *Application user* integrates a particular version of a model into a dataflow to compute the predictions. There are two approaches for obtaining model inferences. In a *push* model, an inference is computed only when a model inference is requested. Alternatively, a model may run in a *pull* model. In the latter mode, a process continuously invokes a model in a windowed input and captures its inferences, passing them over to an output consumption mechanism. Finally, the performance of a model is continuously monitored, and a flag is set once the metrics signal a performance deterioration. At this point, some response has to be developed to avoid providing erroneous predictions.

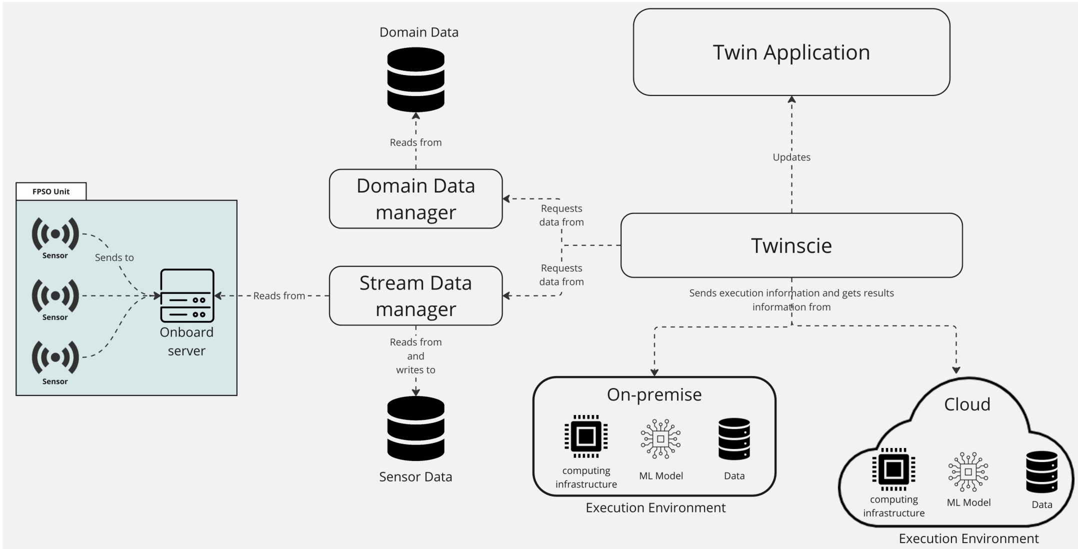

Figure 1: Digital Twin System Overview

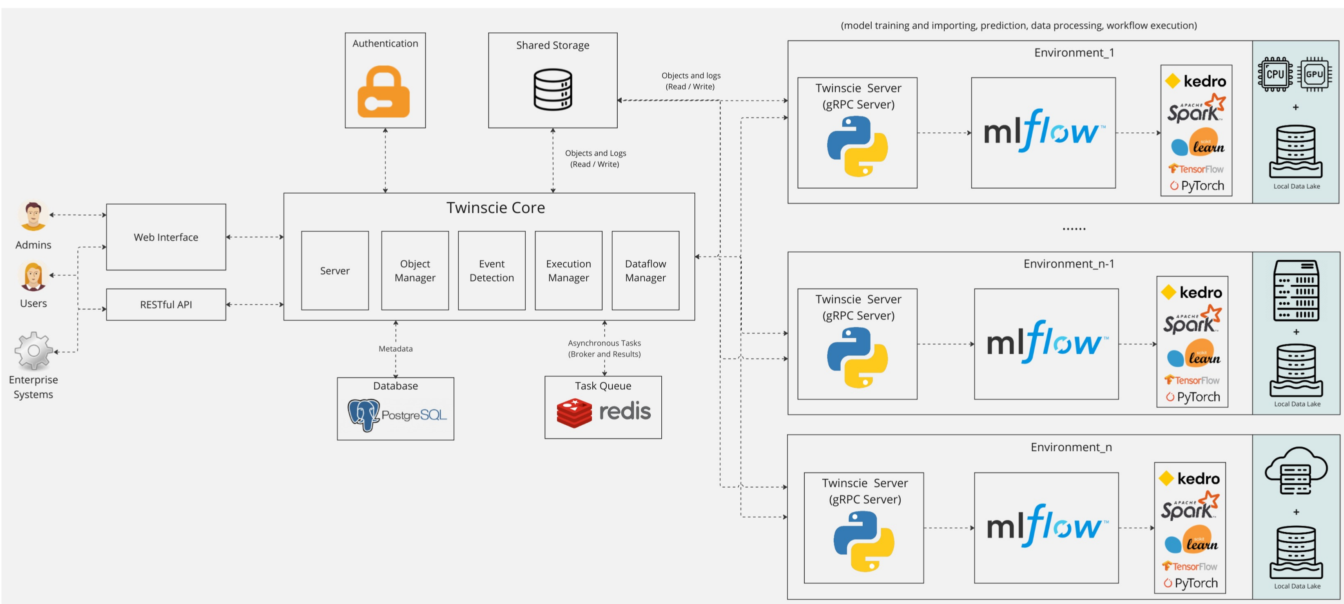

Figure 2: Overview of Twinscie System Components

4.3 Data Management

Twinscie data management comprehends the following aspects: dataset files, dataset metadata, dataset interface, model hyperparameters, and provenance information.

Dataset files are data artifacts physically stored in persistent media. Datasets need to be registered into the system to be referred to. An Application or an Administration user can exercise registration through the system web interface or by an application through the system API. One can register a dataset in any storage structure available in one of the known computational environments. Each dataset file receives a logical id (i.e., a URI), uniquely identifying it within the system. Datasets have metadata associated with them, including the application domain in which they are taken, their size, and their type.

A dataset interface specifies an in-memory dataset and can be specialized for the different in-memory structures a dataflow may implement. This includes distributed in-memory data structures like Spark RDD or a Pandas DataFrame. Another important type of data managed by Twinscie is model hyperparameters. These hyperparameters assist Model Developers and Application Users in tuning the parameters for the available models.

Finally, we store provenance information regarding functions and dataflow executions. This includes dataset files read and written by dataflows and those imported from external systems.

4.4 Dataflow Language and Execution

Data transformations and the models' execution processes are modeled in Twinscie as dataflows. A Dataflow definition models a complete data process through a data dependency graph that specifies the producer-consumer relationship between independent activities. In Twinscie, activities are Python functions. A user with the administrator role can register a Python script with a Python function as an entry point in Twinscie that can be referred to as an activity in a dataflow. Once the functions of interest for a dataflow have been registered, an abstract dataflow can be specified as a directed acyclic graph (DAG), where vertices are either a set of functions, as described above, or a placeholder for files, and edges represent the producer-consumer relationship between functions.

The idea of having an abstract representation is to foster the reuse of the dataflow and to enable its mapping to different dataflow languages and engines. For example, a dataflow specified to run on a small-to-medium-size dataset may be mapped to a desktop environment running a *Kedro* dataflow engine [1], while its invocation on large datasets requires scheduling it on a Big Data platform, such as *Apache Spark* [2]. A concrete dataflow is produced by mapping the directed graph dataflow definition onto a dataflow language and substituting the file placeholders with datasets. Currently, the system maps the *Kedro* dataflow engine. Input datasets correspond to Twinscie registered datasets, whereas a logical address and a filename specify output datasets. Once a concrete dataflow has been instantiated, it can be scheduled for execution in one of Twinscie's known environments. Twinscie stores provenance data about the functions that have been run and the files consumed and produced by running a dataflow.

4.5 Interface with the running environment

The execution of Twinscie dataflows relies on the services provided by the *MLflow* system [3] [44]. Each Twinscie environment runs a component called Twinscie Server alongside an instance of *MLflow*. The Twinscie Server is a gRPC server responsible for receiving execution requests and delegating them to the MLflow instance, encapsulating the task execution.

All Twinscie tasks can be modeled as a dataflow, although specialized API functions exist for individual task invocation. While executing a Twinscie dataflow, *MLflow* stores artifacts, metrics, and parameter values used or generated by the dataflow. Twinscie collects these logs and stores them in its catalog. To prevent each dataflow function or learner from specifying what should be registered to *MLflow*, its auto-logging feature automatically captures metrics, parameters, and models without needing explicit log statements in the dataflow code.

[1]https://docs.kedro.org/
[2]https://spark.apache.org/
[3]https://mlflow.org

Another feature of *MLFlow* is its capability to package dataflows along with their dependencies, creating isolated virtual environments that include the necessary runtimes and libraries for execution. This is especially beneficial because it allows multiple dataflows with varying dependencies and component versions to run simultaneously without impacting other instances.

MLflow also offers other valuable features, such as model serving and versioning. These capabilities are not included in the current version of Twinscie but will be evaluated for future incorporation.

5 STREAM AND DOMAIN DATA MANAGER

The Stream and Domain Data Managers are the components responsible for accessing the different data sources, extracting data, and preparing them to be accessed by the Twinscie system. The domain data corresponds to the non-real-time data, including domain objects' characteristics and contextual information, while stream data refers to observational data captured by sensors. Our current implementation for the Stream Data Manager leverages a combination of advanced software components to collect, process, and analyze data in real time. The framework is primarily based on a data streaming system, the *Event Hub*, which serves as the core for ingesting and processing events collected by sensors.

These Data Managers use an API-Server interface to manage interactions with the Twinscie system. This interface receives requests and responds with requested data. A Query Manager component plays a crucial role in controlling the flow of requests and optimizing traffic between the API-Server and the intermediate data store.

To ensure scalability and efficient management of computing resources, the system is orchestrated by *Kubernetes*, which automates the distribution and scaling of containers hosting the various system components. This orchestration improves operational efficiency and ensures high availability and reliability.

6 ML MODELS FOR FPSO MOORING SYSTEMS

In the oil and gas industry, the trend of increasing petroleum exploration in deep and ultra-deep waters has led to a rise in the utilization of moored FPSO platforms in recent years. These units require a sophisticated mooring system to securely maintain their position. A typical mooring system comprises multiple mooring lines, each costing millions of dollars.

The integrity of the mooring lines condition cannot be directly observed by the operation as they are submerged, so the onboard staff may not immediately notice a line failure. Harsh environmental conditions can escalate the tension on the remaining lines, potentially triggering a cascade effect where multiple lines could fail. The failure of several lines poses a substantial risk of environmental damage due to oil leakage. To mitigate this risk, regular inspections of the mooring system are conducted using remote-operated underwater vehicles or, in certain cases, human divers. However, these inspection operations are intricate, hazardous, and expensive regardless of the method employed. To offer a viable alternative to these labor-intensive inspections, ML models have been developed and integrated into digital twins. These models aim to detect potential failures in the mooring system, thus providing a proactive approach to maintenance and risk management.

Four models are been developed and are gradually being integrated into the Digital Twin system. Two models, NeMo and NeRo, monitor and predict the state of mooring systems using data from sensors that detect platform motion and environmental conditions. Another model, NeuroSim, has been designed to simulate these motions, assisting in the optimization of mooring system designs. Finally, the SeSO model has been developed to extract critical information from the company's technical reports that is not captured by sensors.

The real-time monitoring capabilities of the system empower operators to promptly address emerging issues. By integrating the models into the DT platform, a comprehensive overview of platform condition is achieved, facilitating well-informed decision-making and mitigating the risk of accidents. This improvement in operational oversight increases the safety and reliability of platform operations, leading to minimized downtime and reduced repair expenses. Consequently, this advancement contributes to the optimization of oil platform operations, fostering greater efficiency and sustainability in the long run. In the following subsections, we will provide detailed descriptions of each of these four models.

6.1 NeMo: Neural Motion Estimator

NeMo detects mooring line breaks by identifying changes in relation to the expected motion of the FPSO [15]. To do this, NeMo observes a window of data relating to the six degrees of freedom of the platform's motions and predicts the expected future motion for the intact platform. The six degrees of freedom refer to the possible motions of a rigid body in three-dimensional space, such as an FPSO platform at sea. These movements are divided into three translations (surge, sway, heave) and three rotations (roll, pitch, heading). The horizontal translation motions can also be referred in an inertial system such as UTM - universal transverse Mercator. UTMN referrers to the north-south motion and UTME referrers to east-west motion. These six degrees of freedom allow for a complete description of a vessel's movements in a three-dimensional environment, accounting for both changes in position and changes in orientation. If the predicted motion differs from the sensed motion, an alert is generated. NeMo can handle both data generated by a numerical simulator and data captured by sensors installed on the platforms. Its architecture combines Recurrent Neural Networks (RNNs) with a Graph Neural Network (GNN) [13, 32], being grounded on three primary principles:

- **Time Encoding Using Periodic Functions:** The instants in which time series data are captured are encoded with periodic functions and aggregated to the data, making it possible to capture periodic patterns in the series.
- **Independent Encoding of Time Series:** Independent RNNs are used to process each data stream (including the respective encoded time) [36], ensuring effective operation of the model for different dynamics even if one or more sensors fail.
- **Information Diffusion with GNN:** Each encoded time series is associated with a node in a GNN [40, 42] that facilitates the dissemination of information among its nodes, enriching each representation based on neighboring information. This enriched information is then decoded to predict future platform motions.

The ability to operate with incomplete data makes NeMo a robust tool for proactive management of mooring systems, demonstrating good results in preliminary tests [15].

6.2 NeRo: Natural Frequency Regression and Estimation

Another AI model that detects mooring line breaks is NeRo. It is based on the hypothesis that the resonant frequency of the platform's horizontal motions changes when one or more mooring lines fail. The NeRo model comprises three main modules:

- **Estimator:** Calculates the resonant frequency and damping of the platform based on its actual motion data.
- **Regressor:** Utilizes the CatBoost [28] algorithm to estimate the platform's resonant frequency under normal conditions. It uses the platform's displacement, draft, damping, signal maximum, and minimum to make these estimates.
- **Classifier:** Combines outputs from the Estimator and Regressor to indicate the mooring system's status. Provides several outputs: one for the probability of the mooring system being intact, and others for the probability of failure in each group of lines.

NeRo's strategy showed good separation between "intact" and "broken" data. This approach allows for high-precision identification of mooring line failures, although it is sensitive to untrained modifications in the mooring system, potentially leading to false positives.

6.3 NeuroSim: Neural Simulator

Unlike the two previous models, NeuroSim is a simulator for motion statistics of floating units based on AI Neural Networks (NNs). Its main applications are:

- **Mooring System Design:** NeuroSim can be trained on high-precision simulations of critical centennial metocean conditions. After initial training, FPSO motion statistics are provided almost instantly for new incoming metocean conditions, facilitating the evaluation of numerous conditions during the design phase.
- **Monitoring Operational Floating Units:** NeuroSim maps incident environmental conditions into FPSO motion statistics, providing an additional safety layer. Its meta-models are continuously trained with new measurements, acting as an online predictor of vessel motion statistics.

NeuroSim comprises two meta-models specialized in predicting different motion statistics. It receives input variables associated with environmental conditions and draft. The meta-models predict statistics such as Maximum Roll, Standard Deviation of Roll, and projections in the North and East of the Maximum and Average Offset of the FPSO's Center of Gravity .

6.4 SeSO: Semantic Search for Offshore Engineering

The SeSO model has the potential to evolve into a vital component of DT architectures. Built using a Retriever-Ranker-Reader architecture proposed by Nogueira and Cho [25], SeSO operates by retrieving and ranking documents based on user queries and extracting relevant information to provide precise answers. This robust pipeline ensures that SeSO returns the most relevant information from a large collection of documents, enhancing the accuracy and reliability of the answers.

SeSO could seamlessly interact with digital twins as a communication interface, translating complex data into understandable and actionable insights for users. The process begins when a user inputs a question; the Retriever module returns 100 documents that best match the query. The Ranker module reclassifies these documents and delivers the top 10 to the Reader, which extracts and presents the answers. This process ensures that users receive the most relevant and accurate information, making SeSO a powerful tool for querying data within a DT platform. By integrating SeSO, operators could easily access specific data points or trends from the digital twin, facilitating informed decision-making and operational efficiency.

Moreover, SeSO can function as an analysis assistant within DT platforms, leveraging its advanced natural language processing capabilities to interpret and contextualize sensor data. For instance, SeSO could analyze patterns in mooring line tensions or environmental conditions, providing predictive insights and recommendations for preventive actions. This integration would enable real-time monitoring and proactive management of offshore platforms. Additionally, the availability of domain-specific created language models, further enhances SeSO's ability to perform tasks beyond question-answering, contributing to various natural language processing applications within the DT framework. Combining SeSO's question-answering framework with real-time data analysis from digital twins could drive operational safety, efficiency, and sustainability advancements in Offshore Engineering.

7 USE CASE: NEMO

Key features of the proposed DT platform have been developed, with NeMo being the first model deployed to the Twinscie component. Figure 3 illustrates the execution process of NeMo within the platform.

The process we are implementing begins with an operator inputting data from the Dynasim simulator [24]. This entails handling 168 GB of data spread across 98,000 files for a single platform. Twinscie then executes the preprocessing and training of this data to produce a model, which subsequently undergoes expert validation. In the next step, historical data (currently 5.6 GB) is inputted, preprocessed, and used to fine-tune the model generated with simulated data, to then create a refined model with measured data. This refined model undergoes further validation by an expert. The inference process operates periodically in 6-hour windows, and uses the refined model to act as a digital sensor to detect deviations from expected platform behavior. These deviations indicate potential issues that may require intervention or further investigation.

An example of NeMo results is presented in Figure 4. It shows the Roll and UTME ground truth plot and simulated data with a 95% confidence interval band in a fixed time window. The training was conducted on a platform under 800 different environmental conditions. These conditions were derived from the Hindcast ERA5 dataset for 2021 and 2022, along with data from the Copernicus Marine Environment Monitoring Service (CMEMS). The training process considered five different drafts, ranging from 16 to 20 meters, assuming the mooring lines were intact. The model was trained on 24,000 hours of simulated data (due to confidentiality), with the neural network training lasting for 50 hours. For the inference phase using simulated data, NeMo was tested under 200 environmental conditions from the ERA5 dataset. Similar to the training phase, this phase considered five drafts ranging from 16 to 20 meters, with the remaining mooring lines intact. The inference was conducted on 6,000 hours of simulated data. This visualization highlights NeMo's accuracy in predicting platform behavior, demonstrating the model's robustness under varying conditions.

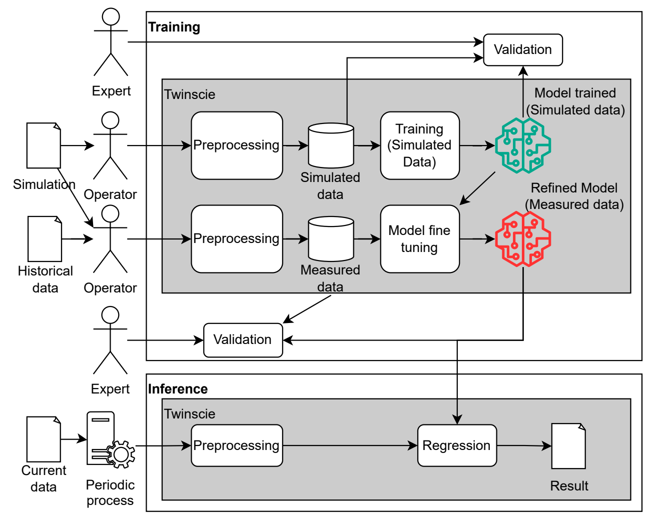

Figure 3: Execution of NeMo in Twinscie.

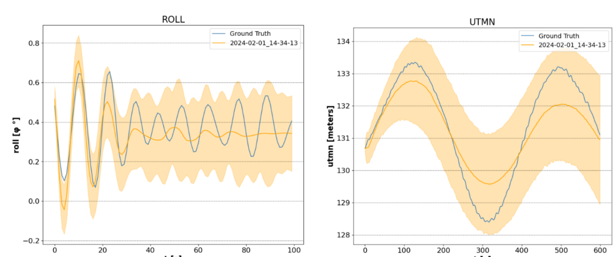

Figure 4: Roll and UTME ground truth and simulated data with a 95% confidence interval band in a fixed time window. An alert is generated whenever the predicted motion goes beyond the confidence interval learned in the training phase.

8 CONCLUSION

Digital Twins supports business decisions and real-time monitoring of systems and processes through their digital representation and object state synchronization. They leverage a powerful combination of new hardware, cloud, supercomputer systems, machine learning models, and abundant data, both in-stream and in-static modes, to achieve accurate digital approximations of the mirrored system.

However, implementing DT systems is challenging as it requires integrating components to support different stages of a twin system. This paper presents an ongoing effort to implement a Digital Twin for the oil and gas industry. We exemplify its adoption in supporting the integrity monitoring system of FPSO platform mooring lines.

We highlight the Twinscie system, a DT component that manages the complete machine learning and data life cycles, and support the Twin Application component. Moreover, we describe the models NeMo, NeRo, NeuroSim, and SeSO that apply advanced machine learning techniques to detect mooring system failures, simulate platform motions, and answer technical questions. Integrating these technologies with DT allows real-time monitoring and proactive management, ensuring safer and more efficient operations. The robust architecture and innovative principles of these models provide powerful tools for maintaining the structural and operational integrity of FPSO platforms, contributing to more sustainable operations in the oil and gas industry.

We conducted tests of our DT platform using the NeMo model in Petrobras's on-premises environment. The tests demonstrated not only NeMo's capability to accurately predict mooring line failures by analyzing platform motion data, but also the value of the entire proposed DT platform in facilitating training, deployment, and continuous inference. These results have provided valuable insights that have significantly improved the overall system. While these initial tests focused on NeMo, we plan to enhance the platform by integrating additional capabilities and testing the other models, NeRo, NeuroSim, and SeSO, to further validate and expand the system's functionalities.

In the future we intend to design and implement automatic decision-making within the system components. We plan to automatically select the best candidate models for a given inference. We also aim to identify concept drifts in the input data and flag models for updates. Additionally, given the different available execution environments, we intend to integrate a cost-based pipeline scheduling model into the pipeline execution process.

ACKNOWLEDGMENTS

The authors thank CAPES, FAPERJ, CNPq, Petrobras, and ANP for partially sponsoring this paper.

REFERENCES

[1] 2019. *Developing a Digital Twin: The Roadmap for Oil and Gas Optimization.* SPE Offshore Europe Conference and Exhibition, Vol. Day 1 Tue, September 03, 2019.

[2] 2020. *Optimizing Drilling Wells and Increasing the Operation Efficiency Using Digital Twin Technology.* SPE/IADC Drilling Conference and Exhibition, Vol. Day 1 Tue, March 03, 2020. https://doi.org/10.2118/199566-MS

[3] 2021. *A Concept for Floating Offshore Wind Mooring System Integrity Management Based on Monitoring, Digital Twin and Control Technologies.* International Conference on Offshore Mechanics and Arctic Engineering, Vol. Volume 9: Ocean Renewable Energy. https://doi.org/10.1115/OMAE2021-61936

[4] 2022. *Digital Twin and Big Data Technologies Benefit Oilfield Management.* Abu Dhabi International Petroleum Exhibition and Conference, Vol. Day 3 Wed, November 02, 2022. https://doi.org/10.2118/211116-MS

[5] 2022. *Enhancing Drilling and Production Through a Digital Twin Methodology.* Abu Dhabi International Petroleum Exhibition and Conference, Vol. Day 2 Tue, November 01, 2022. https://doi.org/10.2118/211120-MS

[6] 2023. *Digital Twin in Oil and Gas Equipment for Predictive Maintenance.* Offshore Technology Conference Brasil, Vol. Day 2 Wed, October 25, 2023. https://doi.org/10.4043/32857-MS

[7] 2023. *Mooring Line Failure Detection Using Artificial Neural Networks: An Application to Field Data Including Artificial Failure Cases.* OTC Offshore Technology Conference, Vol. Day 3 Wed, May 03, 2023. https://doi.org/10.4043/32181-MS

[8] 2023. *NeMo: A Multi-Draft Mooring Line Failure Detection System Based on FPSO Predicted Motion.* International Conference on Offshore Mechanics and Arctic Engineering, Vol. Volume 1: Offshore Technology. https://doi.org/10.1115/OMAE2023-104351

[9] Mohsen Attaran and Bilge Gokhan Celik. 2023. Digital Twin: Benefits, use cases, challenges, and opportunities. *Decision Analytics Journal* 6 (2023), 100165. https://doi.org/10.1016/j.dajour.2023.100165

[10] Sharmin Attaran, Mohsen Attaran, and Bilge Gokhan Celik. 2024. Digital Twins and Industrial Internet of Things: Uncovering operational intelligence in industry 4.0. *Decision Analytics Journal* 10 (2024), 100398. https://doi.org/10.1016/j.dajour.2024.100398

[11] Diego Barboza, Wesley Oliveira, Marco Saraiva, and Leonildes Soares de Melo Filho. 2019. DEMO: Virtual Reality Digital Twin for Floating Production Storage and Offloading (FPSO) Units. 31–32. https://doi.org/10.5753/svr_estendido.2019.8463

[12] Claudio Barros. 2024. *Foundational Framework for Digital Twins: Unifying Mathematical, Ontological, and Computational Perspectives.* Ph. D. Dissertation. National Laboratory of Scientific Computing.

[13] Ben Chamberlain, James Rowbottom, Maria I. Gorinova, Michael M. Bronstein, Stefan Webb, and Emanuele Rossi. 2021. GRAND: Graph Neural Diffusion. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 1407–1418. http://proceedings.mlr.press/v139/chamberlain21a.html

[14] Daniel N. Ramos da Silva, Adolfo Simões, Carlos Cardoso, Douglas E. M. de Oliveira, João N. Rittmeyer, Klaus Wehmuth, Hermano Lustosa, Rafael S. Pereira, Yania Molina Souto, Luciana E. G. Vignoli, Rebecca Salles, Heleno de S. C. Jr, Artur Ziviani, Eduardo S. Ogasawara, Flávia Coimbra Delicato, Paulo de Figueiredo Pires, Hardy L. da C. Pereira Pinto, Luciano Maia, and Fábio Porto. 2019. A Conceptual Vision Toward the Management of Machine Learning Models. In *Proceedings of the ER Forum and Poster & Demos Session 2019 on Publishing Papers with CEUR-WS co-located with 38th International Conference on Conceptual Modeling (ER 2019), Salvador, Brazil, November 4, 2019 (CEUR Workshop Proceedings, Vol. 2469)*, José Ignacio Panach, Renata S. S. Guizzardi, and Daniela Barreiro Claro (Eds.). CEUR-WS.org, 15–27. https://ceur-ws.org/Vol-2469/ERForum2.pdf

[15] Marcel Rodrigues de Barros, Thiago Lizier Rissi, Eduardo Faria Cabrera, Eduardo Aoun Tannuri, Edson Satoshi Gomi, Rodrigo Augusto Barreira, and Anna Helena Reali Costa. 2023. Embracing Data Irregularities in Multivariate Time Series with Recurrent and Graph Neural Networks. In *Intelligent Systems*, Murilo C. Naldi and Reinaldo A. C. Bianchi (Eds.). Springer Nature Switzerland, Cham, 3–17.

[16] Abigail Francisco, Neda Mohammadi, and John E. Taylor. 2020. Smart City Digital Twin–Enabled Energy Management: Toward Real-Time Urban Building Energy Benchmarking. *Journal of Management in Engineering* 36, 2 (2020), 04019045. https://doi.org/10.1061/(ASCE)ME.1943-5479.0000741

[17] Abid Haleem, Mohd Javaid, Ravi Pratap Singh, and Rajiv Suman. 2023. Exploring the revolution in healthcare systems through the applications of digital twin technology. *Biomedical Technology* 4 (2023), 28–38. https://doi.org/10.1016/j.bmt.2023.02.001

[18] Sohaib Ali Khan, Hafiz Zia Ur Rehman, Asad Waqar, Zeashan Hameed Khan, Muntazir Hussain, and Usman Masud. 2023. Digital Twin for Advanced Automation of Future Smart Grid. In *2023 1st International Conference on Advanced Innovations in Smart Cities (ICAISC)*. 1–6. https://doi.org/10.1109/ICAISC56366.2023.10085428

[19] Tarwan Kumar Khatri, Manzoor Ahmed Hashmani, Hasmi Taib, Nasir Abdullah, and Lukman Ab. Rahim. 2023. A Deep Learning-Based Approach to Failure Detection in Mooring (Thin) Lines from Marine Images. *Engineering Proceedings* 56, 1 (2023). https://doi.org/10.3390/ASEC2023-15926

[20] Francisco Paiva Knebel, Rafael Trevisan, Givanildo Santana do Nascimento, Mara Abel, and Juliano Araujo Wickboldt. 2023. A Study on Cloud and Edge Computing for the Implementation of Digital Twins in the Oil & Gas Industries. *Computers & Industrial Engineering* 182 (2023), 109363. https://doi.org/10.1016/j.cie.2023.109363

[21] Mengnan Liu, Shuiliang Fang, Huiyue Dong, and Cunzhi Xu. 2021. Review of Digital Twin about Concepts, Technologies, and Industrial Applications. *Journal of Manufacturing Systems* 58 (2021), 346–361.

[22] Xuan Liu, He Gan, Ying Luo, Yangquan Chen, and Liang Gao. 2023. Digital-Twin-Based Real-Time Optimization for a Fractional Order Controller for Industrial Robots. *Fractal and Fractional* 7, 2 (2023). https://doi.org/10.3390/fractalfract7020167

[23] Stefan Mihai, Mahnoor Yaqoob, Dang V Hung, William Davis, Praveer Towakel, Mohsin Raza, Mehmet Karamanoglu, Balbir Barn, Dattaprasad Shetve, Raja V Prasad, et al. 2022. Digital twins: A survey on enabling technologies, challenges, trends and future prospects. *IEEE Communications Surveys & Tutorials* 24, 4 (2022), 2255–2291.

[24] Kazuo Nishimoto, Carlos H. Fucatu, and Isaias Q. Masetti. 2002. Dynasim—A Time Domain Simulator of Anchored FPSO . *J. Offshore Mech. Arct. Eng.* 124, 4 (10 2002), 203–211. https://doi.org/10.1115/1.1513176

[25] Rodrigo F. Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *CoRR* abs/1901.04085 (2019). arXiv:1901.04085 http://arxiv.org/abs/1901.04085
[26] Rafael S. Pereira, Yania Molina Souto, Anderson Chaves da Silva, Rocío Zorilla, Brian Tsan, Florin Rusu, Eduardo S. Ogasawara, Artur Ziviani, and Fábio Porto. 2021. DJEnsemble: a Cost-Based Selection and Allocation of a Disjoint Ensemble of Spatio-temporal Models. In *SSDBM 2021: 33rd International Conference on Scientific and Statistical Database Management, Tampa, FL, USA, July 6-7, 2021*, Qiang Zhu, Xingquan Zhu, Yicheng Tu, Zichen Xu, and Anand Kumar (Eds.). ACM, 226–231. https://doi.org/10.1145/3468791.3468806
[27] Fabio Porto and Patrick Valduriez. 2022. Data and Machine Learning Model Management with Gypscie. In *CARLA 2022 - Workshop on HPC and Data Sciences meet Scientific Computing*, Advanced Computing System for Latin America and the Caribbean (SCALAC) (Eds.). SCALAC, Porto Alegre, Brazil, 1–2. https://hal-lirmm.ccsd.cnrs.fr/lirmm-03799097
[28] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. CatBoost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf
[29] Guilherme Rangel, Giovanni do Amaral, Kennedy Leandro, Raul Dotta, Guilherme Franzini, Alfredo Neto, Edgard Malta, Luís Bitencourt Jr, and Guilherme Martins. 2021. Methodology for Obtaining a Digital Twin for a FPSO Mooring System. https://doi.org/10.26678/ABCM.COBEM2021.COB2021-0749
[30] Hamed Rezaniaiee Aqdam, Mir Mohammad Ettefagh, and Reza Hassannejad. 2018. Health monitoring of mooring lines in floating structures using artificial neural networks. *Ocean Engineering* 164 (2018), 284–297. https://doi.org/10.1016/j.oceaneng.2018.06.056
[31] Vítor Ribeiro, Eduardo H. M. Pena, Raphael de Freitas Saldanha, Reza Akbarinia, Patrick Valduriez, Falaah Arif Khan, Julia Stoyanovich, and Fábio Porto. 2023. Subset Modelling: A Domain Partitioning Strategy for Data-efficient Machine-Learning. In *Proceedings of the 38th Brazilian Symposium on Databases, SBBD 2023, Belo Horizonte, MG, Brazil, September 25-29, 2023*. SBC, 318–323. https://sol.sbc.org.br/index.php/sbbd/article/view/25537
[32] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The Graph Neural Network Model. *IEEE Trans. Neural Networks* 20, 1 (2009), 61–80. https://doi.org/10.1109/TNN.2008.2005605
[33] Marius Schlegel and Kai-Uwe Sattler. 2023. Management of Machine Learning Lifecycle Artifacts: A Survey. *SIGMOD Rec.* 51, 4 (jan 2023), 18–35. https://doi.org/10.1145/3582302.3582306
[34] Marius Schlegel and Kai-Uwe Sattler. 2023. Management of Machine Learning Lifecycle Artifacts: A Survey. *SIGMOD Rec.* 51, 4 (jan 2023), 18–35. https://doi.org/10.1145/3582302.3582306
[35] Fei Shen, Shuang Shuang Ren, Xiang Yang Zhang, Hong Wu Luo, and Chao Min Feng. 2021. A Digital Twin-Based Approach for Optimization and Prediction of Oil and Gas Production. *Mathematical Problems in Engineering* 2021, 1 (2021), 3062841. https://doi.org/10.1155/2021/3062841
[36] Corentin Tallec and Yann Ollivier. 2018. Can recurrent neural networks warp time?. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=SJcKhk-Ab
[37] Fei Tao, He Zhang, Ang Liu, and A. Y. C. Nee. 2019. Digital Twin in Industry: State-of-the-Art. *IEEE Transactions on Industrial Informatics* 15, 4 (2019), 2405–2415. https://doi.org/10.1109/TII.2018.2873186
[38] Alexandre Vallée. 2024. Envisioning the Future of Personalized Medicine: Role and Realities of Digital Twins. *J Med Internet Res* 26 (13 May 2024), e50204. https://doi.org/10.2196/50204
[39] Eric VanDerHorn and Sankaran Mahadevan. 2021. Digital Twin: Generalization, Characterization and Implementation. *Decision Support Systems* 145 (2021), 113524.
[40] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph Attention Networks. *CoRR* abs/1710.10903 (2017). arXiv:1710.10903 http://arxiv.org/abs/1710.10903
[41] Thumeera R. Wanasinghe, Leah Wroblewski, Bui K. Petersen, Raymond G. Gosine, Lesley Anne James, Oscar De Silva, George K. I. Mann, and Peter J. Warrian. 2020. Digital Twin for the Oil and Gas Industry: Overview, Research Trends, Opportunities, and Challenges. *IEEE Access* 8 (2020), 104175–104197. https://doi.org/10.1109/ACCESS.2020.2998723
[42] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Peng Cui, Philip S. Yu, and Yanfang Ye. 2019. Heterogeneous Graph Attention Network. *CoRR* abs/1903.07293 (2019). arXiv:1903.07293 http://arxiv.org/abs/1903.07293
[43] Yajuan Xie and Hesheng Tang. 2024. A deep learning approach based on domain generalization for mooring tension prediction in floating structures. *Ocean Engineering* 306 (2024), 118128. https://doi.org/10.1016/j.oceaneng.2024.118128
[44] Matei A. Zaharia, Andrew Chen, Aaron Davidson, Ali Ghodsi, Sue Ann Hong, Andy Konwinski, Siddharth Murching, Tomas Nykodym, Paul Ogilvie, Mani Parkhe, Fen Xie, and Corey Zumar. 2018. Accelerating the Machine Learning Lifecycle with MLflow. *IEEE Data Eng. Bull.* 41 (2018), 39–45. https://api.semanticscholar.org/CorpusID:83459546