



Reference architecture for digital twin-based predictive maintenance systems

Raymon van Dinter^{a,b}, Bedir Tekinerdogan^{b,*}, Cagatay Catal^c

^a Sioux Technologies, Apeldoorn, the Netherlands

^b Information Technology Group, Wageningen University & Research, Wageningen, the Netherlands

^c Department of Computer Science and Engineering, Qatar University, Doha, Qatar

ARTICLE INFO

Keywords:

Reference architecture
Design
System architecture
Framework
Digital twin
Predictive maintenance

ABSTRACT

Context: Digital Twin-based predictive maintenance systems are frequently integrated into complex systems. The success of the integration depends on the design of the system. A Reference Architecture can be used as a blueprint to design Application Architectures rapidly and consistently for various application domains, resulting in a reduced time-to-market.

Objective: The main objective of this study is to develop and evaluate a Reference Architecture designed using renowned software architecture methods.

Method: A domain analysis was performed to gather and synthesize the literature on Digital Twin-based predictive maintenance systems, which we used to model the key features. We applied UML diagrams to design the reference architecture based on the feature model. We evaluated the reference architecture using three case studies.

Results: We derived three views for Digital Twin-based predictive maintenance systems. For the user's view, we developed a context diagram. We developed a package diagram for the structural view, and we designed a layered view to show the system's decomposition in layers. We designed an Application Architecture for each case study based on the study's features using each Reference Architecture view. Additionally, we designed a deployment view to describe the hardware and software and its environment.

Conclusion: We demonstrated that the methods of creating a Reference Architecture could be used in the Digital Twin-based predictive maintenance domain and showed how an Application Architecture could be designed in this context.

1. Introduction

Since the industrial revolution, machines have been used increasingly in all domains, improving product performance, efficiency, and the consistency of quality. However, machine condition greatly affects business models dependent on machines. Until recently, machines were usually repaired after failure or maintained periodically. However, these two methods have a few downsides. First, the risk of breakdown is not minimized; as such, there is still a risk of downtime and catastrophic failures. Second, these two methods do not make optimal use of system maintenance engineers, a scarce resource. Third, these methods waste natural resources, as reactive maintenance can damage more than the broken component, and preventive maintenance does not use the full lifetime of a component (Errandonea, Beltrán, & Arizabalaga, 2020).

One method to use the full lifetime of a component is predictive maintenance (PdM), a subgroup of Prognostics and Health Management (PHM). With PdM, we use large datasets of run-to-failure data from sensor-equipped machines. The data can determine several health aspects, such as Remaining Useful Life (RUL), a specific health indicator, or anomaly detection (van Dinter, Tekinerdogan, & Catal, 2022). However, current machines have become well-designed, meaning there is a low amount of historical failure data (Luo, Hu, Ye, Zhang, & Wei, 2020).

Current trends toward improving PdM models are leveraging a Digital Twin (DT) to generate synthetic data. A DT is a high-fidelity digital model of the physical asset based on physics, data, rules, behavior, or geometry (Semeraro, Lezoche, Panetto, & Dassisti, 2021), which acquires sensor data during the operation of equipment and stores

* Corresponding author.

E-mail addresses: raymon.vandinter@wur.nl (R. van Dinter), bedir.tekinerdogan@wur.nl (B. Tekinerdogan), ccatal@qu.edu.qa (C. Catal).

<https://doi.org/10.1016/j.cie.2023.109099>

Received 19 August 2022; Received in revised form 2 February 2023; Accepted 12 February 2023

Available online 16 February 2023

0360-8352/© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

the historical running data for further analysis (Luo et al., 2020). Meanwhile, a DT can generate reliable synthetic data for a more robust predictive maintenance model. A DT-based PdM system comprises two modeling layers: an asset representation layer and an application layer.

DT-based PdM systems are complex and contain many software modules (Luo et al., 2020). The software architecture is crucial for the success of a DT-based PdM system design. Software architecture should describe communication, design decisions, architectural concerns, and architectural decisions addressing these concerns along with the rationale (Tummers, Kassahun, & Tekinerdogan, 2021; Tummers, Tobi, Catal, & Tekinerdogan, 2021). We use the Reference Architecture (RA) design principles with structured and standardized design methods. An RA is a generic, domain-independent design that derives an Application Architecture (AA) based on system requirements. An RA can be considered as a template solution for a particular application domain and it also provides a vocabulary for implementation purposes. Application architecture selects some of the reference architecture elements and should be considered a specialization of the reference architecture. Application-specific changes are introduced in the application architecture. Using an RA enables software architects to develop a consistent design within a smaller timeframe, eventually reducing the time to market for the product. We propose an RA for DT-based PdM systems using well-known software design principles. We use key PdM and DT features from our systematic literature review (van Dinter et al., 2022). To validate our RA, we created three AAs from peer-reviewed literature studies. In addition, the design efforts are reduced, fewer errors occur in the designed system, and the overall quality increases (Ingeno, 2018). Communication within the organization is improved because of having a common architectural mindset. Development costs are reduced because the common assets are reused. The learning curve of developers is improved because of the use of the same vocabulary in the organization. The interoperability of the systems is enhanced by using a standardized solution.

Several studies are related to our work, which will be discussed in this paragraph. Lee, Bagheri, and Kao (2015) proposed the renowned 5C architecture, a layered architecture concept for cyber-physical systems (CPS). As a DT one kind of CPS, we leveraged this study and included the 5C architecture in our RA design. Mihai et al. (2021) designed a DT framework for predictive maintenance in Industry 4.0. Their framework is focused on a data science application using Python and the dashboarding tool Dash. They provide two analysis models that work in parallel: anomaly detection and a remaining useful life estimation module. Nordal and El-Thalji (2021) designed a predictive maintenance system architecture for Industry 4.0 applications called Maintenance 4.0. Their architecture is detailed and can be applied to manufacturing cases. However, they do not include a Digital Twin module in their design. Centomo, Dall'ora, and Fummi (2020) proposes a general framework based on the EDA approach that allows to set up a maintenance strategy by analyzing data retrieved from sensors. The framework is captured in diagrams describing the proposed system's generic behavior and is applied to gearbox degradation monitoring. Cohen and Singer (2021) developed a smart process controller framework for identifying anomalies. They designed an architecture using block diagrams of a typical manufacturing process control system and how to refactor it into a smart process control system. As this study clearly defines the modules used for validation, we use their implementation as a case study. Z. Liu et al. (2018); Z. Liu, Zhang, Xu, Jin, and Lee (2018) designed a CPS architecture for PHM of high-speed railway (HSR) transportation systems. They used the 5C architecture for their framework, specifically creating a layered design for HSR systems. Additionally, they developed a Cyber-Physical Interface Platform for HRS applications. Luo et al. (2020) developed a hybrid predictive maintenance framework based on a DT for CNC machine applications. However, they also provide a multi-domain model implementation. They present a well-structured layered architecture and multiple flowcharts to explain their design choices. Xiong, Wang, Fu, and Xu (2021)

designed a framework and workflow for DT-driven predictive maintenance of aero-engines. Xiong et al. (2021) additionally provides a pipeline from aero-engine sensor data toward remaining useful life predictions.

There are also RAs that concern exclusively predictive maintenance systems or Digital Twins. Even though they are out of scope for this review, we will highlight the main contributions. Uhlmann, Polte, and Koutrakis (2021) developed a holistic concept toward a RA for predictive maintenance systems based on the three-layer Industrial Internet of Things (IIoT), which they validated in one case study applying Machine Learning (ML). Groba, Cech, Rosenthal, and Gossling (2007) proposed an Architecture of a predictive maintenance framework. In their study, in collaboration with SAP research, their architecture is housed as a component between the manufacturing domain and the Enterprise Resource Planning (ERP) systems. Balogh, Gatial, Barbosa, Leitão, and Matejka (2018) propose a RA for predictive maintenance systems using a cloud-based collaborative maintenance services platform for smart manufacturing. Zhidchenko, Startsev, and Handroos (2022) present an RA for interconnected systems running Digital Twins of heavy equipment. They provide a class and sequence diagram to elaborate their data model for Digital Twins. Redelinghuys, Kruger, and Basson (2019) present an RA for manufacturing systems with many Digital Twins that need interaction. They propose the Six-Layer Architecture for Digital Twins with Aggregation (SLADTA). Josifovska, Yigitbas, and Engels (2019) make use of the 5C architecture for CPS. They present a reference framework for the main building blocks of a Digital Twin framework that are missing from the 5C architecture. Lehner, Wolny, Mazak-Huemer, and Wimmer (2020) aim to connect Model-Driven Engineering models created in design time and use the knowledge during runtime. As such, they present a RA that enables data querying from model repositories to add design-time knowledge to the running system and reasoning system states at runtime in design-time models. Aheleroff, Xu, Zhong, and Lu (2021) identified Industry 4.0 technologies and designed a Digital Twin Reference Architecture. They also utilized the Digital Twin as a Service paradigm as a case study for digital transformation in appliances for wetlands schedule maintenance.

The practical contributions of this study are five-fold:

- We developed a feature model using 42 selected articles on predictive maintenance using Digital Twins
- We categorized the features using the 5C architecture for Cyber-Physical Systems
- We proposed and validated a Reference Architecture for Digital Twins-based predictive maintenance systems using triangulation
- We demonstrated the effectiveness of the proposed architecture using three case studies
- We showed that the application architectures can be easily created using the proposed reference architecture.

Our main theoretical contribution is that we added another component to the development of a Reference Architecture (i.e., a graphical model of Predictive Maintenance Systems theory), which is analyzing state-of-the-art literature through a Systematic Literature Review (SLR) methodology. Additionally, we performed a domain analysis on existing Reference Architectures for Predictive Maintenance, Digital Twins, and Digital Twin-based Predictive Maintenance systems. The addition of the systematic literature review and resulting therefrom the feature model provides a structured overview of the mandatory and optional components in the state-of-the-art Digital Twin-based Predictive Maintenance Systems.

The next sections are organized as follows: Section 2 presents the methods. Section 3 explains the results. Section 4 shows the case studies. Section 5 presents the discussion and Section 6 concludes the paper.

2. Methods

2.1. Research questions

To generate a robust RA for DT-based PdM systems, we identified the following research questions:

- RQ1: What are the existing reference architectures for Digital Twin-based predictive maintenance systems?
- RQ2: What is a feasible Reference Architecture for a Digital Twin-based predictive maintenance system?
- RQ3: Does the Reference Architecture allow for the derivation of a specific Application Architecture?

As such, we adopted the RA design approach from [Tummers, Kassahun, & Tekinerdogan, 2021](#), [Tummers, Tobi, et al. \(2021\)](#), shown in [Fig. 1](#). First, we perform a domain analysis, which synthesizes the domain knowledge to support the design of the RA. The domain analysis consists of two activities: domain scoping and modeling. Domain scoping specifies the domain's scope and the knowledge sources required to determine the foundational pillars. The goal of domain modeling is to describe domain knowledge in a consumable manner ([van Geest, Tekinerdogan, & Catal, 2021](#)). In our case, the domain scoping activity consists of a systematic literature review concerning PdM using DTs ([van Dinter et al., 2022](#)), while the domain modeling activity consists of creating a feature model.

Second, we use the information from the feature model for the design of our RA. Therefore, we create several views, which we will elaborate in [Section 3.4](#). To show the suitability of the design, we derive multiple AAs from different application domains. We use a retrospective case study design from three applications, elaborated in the next section. We used the method for deriving the AA from the RR as proposed by [Tummers, Kassahun, & Tekinerdogan, 2021](#), [Tummers, Tobi, et al. \(2021\)](#).

2.2. Case studies

This section describes the selection of case studies. Triangulation is a key principle in the design of a case study. Triangulation means taking multiple viewpoints toward the subject of study to capture the essence of the bigger picture ([Runeson & Höst, 2009](#)). We ensured data triangulation (i.e., various use cases) and observer triangulation (i.e., multiple application domains/observers).

The study was validated using cases involving different PdM requirements in different application domains. As mentioned in the previous section, we used studies implementing PdM algorithms. Then, we

conceptually applied a DT-based PdM system to the case. The following three cases were selected for understanding and validating the RA design:

- Case 1. **New Holland tractor transmission system.** Maintenance of agricultural machinery, including tractors, is routine in the life of any farmer, especially preventive and corrective maintenance. [da Silva, Rodrigues de Sá, and Menegatti \(2019\)](#) created a PdM algorithm that identifies failures in the New Holland T8 tractor transmission system. It is a tractor for large-scale production, and pressure data of the clutch system is used for fault diagnosis.
- Case 2. **Wafer semiconductor manufacturing.** Silicon wafers are extensively employed in the semiconductor industry to fabricate integrated circuits. Typically, wafer thickness is monitored using scan data from a laser probe. Several polishing procedure factors can impact the quality and consistency of the acceptable thickness across the entire wafer. [Cohen and Singer \(2021\)](#) proposed an intelligent supervisory framework for a single process controller suitable for Industry 4.0 shop floors. It is compatible with the paradigm of a Cyber-Physical System (CPS) since its implementation creates a rich cyber-physical entity of the regulated process. This CPS entity might be considered the process's digital twin or a strong foundation for producing it.
- Case 3. **Electro-optical imaging system.** Infrared imaging devices are widely equipped for target recognition and tracking on aircraft and spaceships. Electro-optical detection excels at maintaining stealth and avoiding interference compared to radar and laser detection. However, when the electro-optical system's health deteriorates significantly, the imaging quality suffers, reducing the detection probability to target and, as a result, leading to the failure of combat operations. As a result, [Yu, Song, Tang, and Dai \(2021\)](#) designed a DT model and monitored the electro-optical system's health.

3. Results

This section starts with a summary of related reference architecture studies. It then presents the design phase's four steps: domain analysis, domain scoping, domain modeling, and RA design.

3.1. Domain analysis

To scope and model the domain, we performed a systematic literature review (SLR) ([van Dinter et al., 2022](#)) to gather and synthesize evidence on PdM using DT. We provided an answer to the research

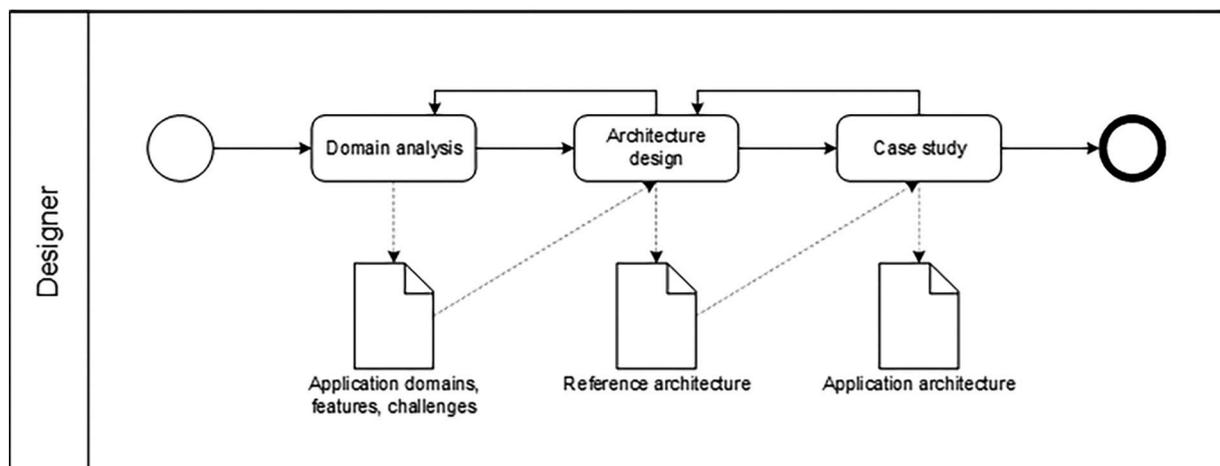


Fig. 1. The methodological design for creating a Reference architecture adopted from [Tummers, Kassahun, & Tekinerdogan, 2021](#), [Tummers, Tobi, et al. \(2021\)](#).

questions as listed in Table 1. We synthesized evidence from 42 selected high-quality articles (Aivaliotis, Arkouli, Georgoulis, & Makris, 2021; Aivaliotis, Georgoulis, Arkouli, & Makris, 2019; Aivaliotis, Georgoulis, & Chryssolouris, 2019; Aivaliotis, Xanthakis, & Sardelis, 2020; Altun & Tavli, 2019; Anis, Taghipour, & Lee, 2020; Barkalov, Dorofeev, Fedorova, & Polovinkina, 2021; Booyse, Wilke, & Heyns, 2020; Cattaneo & MacChi, 2019; Centomo et al., 2020; Cohen & Singer, 2021; Deebak & Al-Turjman, 2021; Desai, Granja, Higgs, & III., 2021; Dhada, Hernández, Palau, & Parlikad, 2021; Heim et al., 2020; Johansen & Nejad, 2019; Kaul, Bender, & Sextro, 2020; Liu, Mauricio, Qi, Peng, & Gryllias, 2020; Z. Liu, Jin, et al., 2018; Z. Liu, Zhang, et al., 2018; Luo et al., 2020; Meraghni et al., 2021; Moghadam & Nejad, 2022; Moghadam, Rebouças, & Nejad, 2021; Moi, Cibicik, & Rølvåg, 2020; Oluwasegun & Jung, 2020; Priyanka, Thangavel, Gao, & Sivakumar, 2021; Qiao, Wang, Ye, & Gao, 2019; Rajesh, Manikandan, Ramshankar, Vishwanathan, & Sathishkumar, 2019; Rossini et al., 2020; Short & Twiddle, 2019; Tao, Zhang, Liu, & Nee, 2018; Tygesen, Worden, Rogers, Manson, & Cross, 2019; Tzanis et al., 2020; Wang, Lee, & Angelica, 2020; Wang, Liu, Liao & Mrad, 2020; Werner, Zimmermann, & Lentjes, 2019; Xiong et al., 2021; Xu, Sun, Liu, & Zheng, 2019; Yu et al., 2021; Zenisek, Wolfartsberger, Sievi, & Affenzeller, 2018; Zhang, Huo, Zheng, & Li, 2020) to answer our research questions. These articles form the set of evidence for our Reference Architecture design. Some of the articles also propose a framework or reference architecture and are, as such, discussed in Section 1 Introduction.

3.2. Domain scoping

DT-based PdM systems cover a wide range of subdomains. Much research is focused on the *Manufacturing* domain. Others focus mainly on the *Energy*, *Aerospace*, and *Automotive* domains. The most common DT abstraction level is the *Component* level and *System* level. Additionally, most DT-based PdM systems apply the *Digital Monitor* design pattern.

3.3. Domain modeling

Feature modeling is a method for representing domain knowledge and features of the system and its interdependencies (van Geest et al., 2021). The feature model is constructed from the information synthesized in the SLR study. A feature model also depicts whether a feature is optional or mandatory. Features mentioned by most publications or features that are required in every system are considered mandatory. Other elements are optional and can be leveraged based on a developer's needs.

We have generated a DT-based PdM system feature model, which has been divided into two figures to establish a robust overview of the

Table 1
The research questions of the literature review (van Dinter et al., 2022).

No.	Research Question (RQ)
RQ1	What is the objective of predictive maintenance using Digital Twins?
RQ2	On which application domains is predictive maintenance using Digital Twins applied?
RQ3	Which Digital Twin platforms are used to develop Digital Twins for predictive maintenance?
RQ4	Which model representation types are used to develop Digital Twins for predictive maintenance?
RQ5	Which approaches are being applied for predictive maintenance using Digital Twins?
RQ6	Which abstraction levels of Digital Twins are used for predictive maintenance?
RQ7	Which Digital Twin design patterns are applied for predictive maintenance?
RQ8	Which communication protocols are used for Digital Twins for predictive maintenance?
RQ9	Which Twinning state parameters are used for predictive maintenance using Digital Twins?
RQ10	What are the challenges and possible solution directions with respect to predictive maintenance using Digital Twins?

features. Its features depend on literature from the SLR study and additional sources regarding data science techniques. Fig. 2 shows the DT-based PdM system with the collapsed PdM feature. Fig. 3 expands the features of the PdM side of the system.

A DT depends on several features, which we address following the order of the 5C architecture as proposed by Lee et al. (2015) (Connection, Conversion, Cyber, Cognition, and Configuration):

Connection. First, the main component of a DT is its physical counterpart. Without the *Physical Twin* (PT), the DT cannot exist. *Data acquisition* is the process of synchronizing input data between the DT and PT systems at a specific Twinning rate. The Twinning rate may be a continuous stream or batches (Giray & Catal, 2021; Tekinerdogan & Verdouw, 2020). *Storage* is a feature supporting other features and refers to data's temporary and persistent storage (Giray & Catal, 2021). The storage feature involves relational and non-relational or NoSQL database systems (Cattaneo & MacChi, 2019; Priyanka et al., 2021; Wang, Lee, & Angelica, 2020). The *communication* feature consists of three sub-features: publish-subscribe, client-server, and primary-secondary, which are the three most common communication architectures.

Conversion. *Security* was not mentioned by most studies identified by the SLR study. However, security is a mandatory feature for industrial applications, as it is undesirable to have an insecure connection between the machine and DT. *Data quality management* refers to handling data quality problems that may arise due to noise and drifting in sensor data, errors, or unreported human interventions. There may be missing, incorrect, unusable, or redundant data (Dhada et al., 2021; Moi et al., 2020; Xiong et al., 2021; Yu et al., 2021).

Cyber. The cyber layer consists of the *simulation* model feature. This feature contains the four DT representation types described by Semeraro et al. (2021): data-driven, behavior, physics-based, and geometrical representation models. A DT can also be represented through hybrid models consisting of two or more models (Luo et al., 2020; Tao et al., 2018; Yu et al., 2021). For example, a physics-based model can describe the known features of an asset, while a data-driven model can estimate the unknown parameters.

Cognition. The cognition layer consists of the *visualization* feature. With this feature, the DT model may be visualized using its FEM model (Wang, Liu, Liao & Mrad, 2020), CAD model (Rajesh et al., 2019), or AR techniques (Deebak & Al-Turjman, 2021). At last, we have the analysis module with *batch* and *stream* processing and *PdM*. The PdM's features are further elaborated using Fig. 3.

Configuration. The configuration layer contains the *machine optimization* feature. Z. Liu, Jin, et al. (2018) describe the machine optimization features with sub-features such as fleet management, zero downtime, optimized performance, and maximizing fleet readiness.

Five new sub-features are shown as we expand the PdM feature in Fig. 3. Again, we base the elaboration structure of the feature model using the Machine Learning workflow, as described by MLOps (2023). This section regards the application modeling layer, which is usually developed using machine learning techniques.

Data Engineering. As the PdM feature model is part of the DT-based PdM feature model, we use the *data acquisition* feature to obtain the necessary data. PdM algorithms may be executed in real-time or periodically, such as once per day. Feature extraction is used to gather more information from the data. It consists of time-domain, frequency-domain, and time-frequency domain feature extraction.

Machine Learning (ML) Model Engineering. As machine failures are not regularly occurring, datasets are often imbalanced. The *data sampling* sub-feature aims to balance the data to enable models to train on the essential features (Lemaître, Nogueira, & Aridas, 2017). When the data has been enriched and balanced, we can split the data needed for the *validation* feature. Engineers can choose between using a holdout set or N*k-fold cross-validation. When the data is correctly split, one must choose the right *evaluation metrics* to measure what it asserts. The *KPI* feature is essential for PdM as it determines the problem to tackle and the type of model to use. For example, Remaining Useful Life (RUL) is a

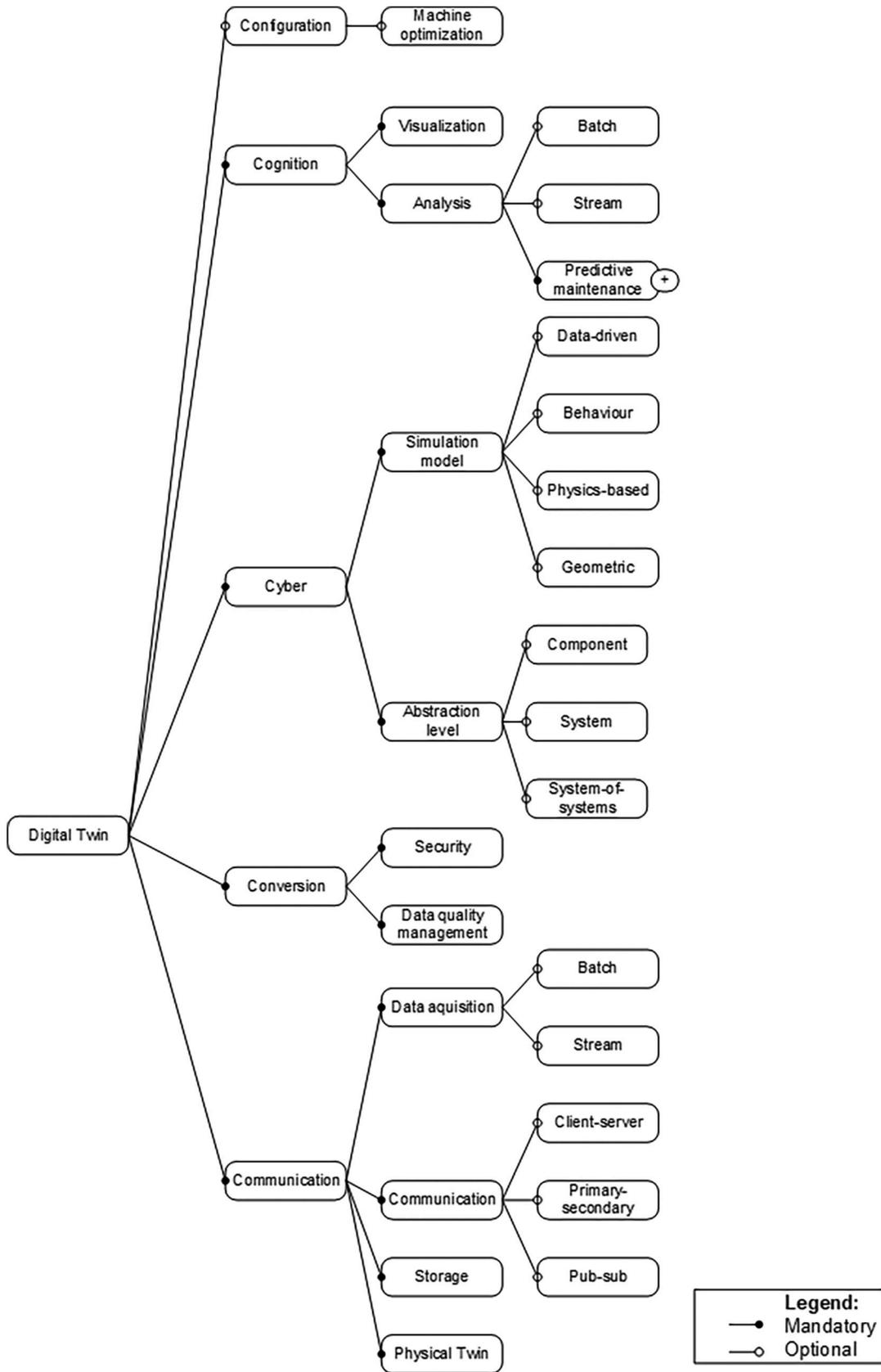


Fig. 2. Feature model of the DT-based PdM system.

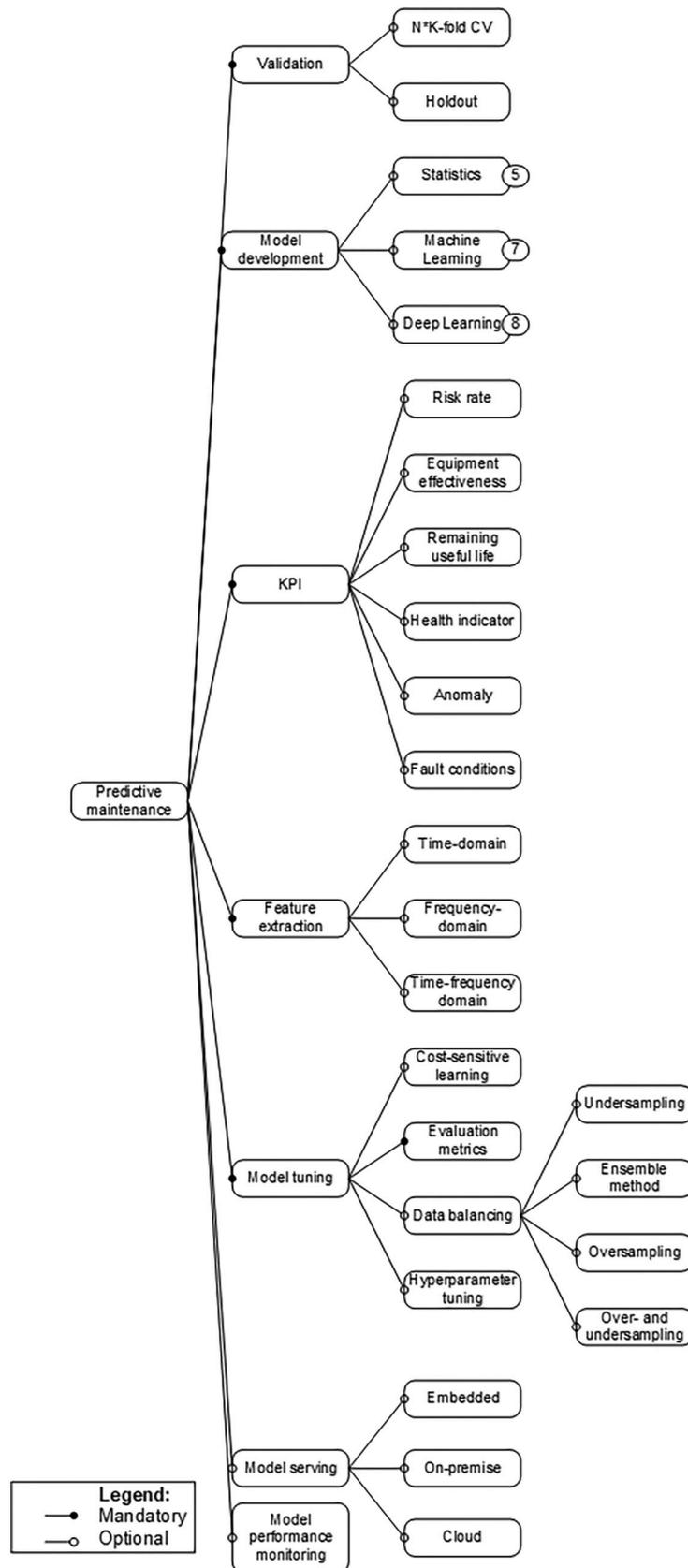


Fig. 3. Expanded feature model of the PdM feature.

regression task, while fault condition is a classification task. Furthermore, the health indicator describes the deterioration of the asset, while RUL describes the time before failure. *Model Development* is the feature where the predictive model is engineered. We identified five types of predictive statistic models from the SLR study, seven types of ML models, and eight types of Deep Learning (DL) models for PdM tasks.

Model deployment. The model deployment step focuses on bringing the model to production (MLOps, 2023). The *model serving* feature enables deployment with three options, as described by Digital Twin Consortium (2023). Optionally, *model performance monitoring* can be added to keep track of model performance in production (MLOps, 2023).

3.4. Reference architecture design

In this section, we describe the selection of *views* and *diagrams* used to describe the RA design and elaborate on each view.

3.4.1. Selection of views

UML diagrams can be separated into five types of views to support the design of a system (Mall, 2018):

- User view
- Structural view
- Layered view
- Behavioral view
- Implementation view

These views map to the UML view models (Alhir, 1998). The relationship between these views is described by Kruchten (1995) as follows: “This use of multiple views allows to address separately the concerns of the various ‘stakeholders’ of the architecture: end-user, developers, systems engineers, project managers, etc., and to handle separately the functional and non-functional requirements” (Kruchten, 1995). We make use of this structure to design the RA. As the Behavior and Implementation views rely on exact system characteristics, we implement the User, Structural, and Layered views. For the User’s view of the system, we develop a level-0 Data Flow Diagram (DFD), also known as a context diagram (Afyenni, 2014). For the Structural view, we provide a package diagram. For the Layered view, we create a layered view. For the case studies, we also develop a deployment diagram for the Implementation view.

3.4.2. User’s view

The DFD level 0, depicted in Fig. 4, is a diagram that describes a system and its interfaces with the external environment at a high level. The diagram shows data sources, such as sensors, and outputs, such as analytic insights, which relate to human supervisors and actuators. As DT systems are often served in the cloud, we introduced cloud computing resources as a bidirectional data source. The data is acquired, stored, processed, and analyzed by the DT-based PdM system to provide information on the DT’s health state.

3.4.3. Structural view

Fig. 5 depicts the package diagram of the DT-based PdM system RA. The *Physical Twin* module is responsible for acting as a Proxy pattern for

communication between the DT and PT. *Data acquisition* modules are responsible for loading data into the DT system for processing. The *Communication* module provides different communication protocols to enable the desired behavior of the system. *Data quality* can be managed by applying cleaning and enrichment procedures to the gathered data to gain more accurate findings from the analysis. The *Abstraction level* module enables the development of the *SimulationModel* in different levels of granularity. The *SimulationModel* module provides several submodules that enable the development of a DT. To help decision-making, submodules can be employed to *Analyze* the data, such as the *Machine optimization* and *Predictive maintenance* modules. The *Batch* and *Stream* modules describe the execution rate of analysis, which may be continuous or periodic. The *Storage* module manages several data storage mechanisms. To secure data from unwanted access, a *Security* module is required.

The *Predictive Maintenance* module is a member of the *Analysis* module. It is responsible for predicting the health of a component or system. It uses the cleaned input data and the *Feature extraction* module to generate relevant features. The *KPI* module provides a selection of key performance indicators that can be predicted. Predictive models are generated in the *Model development* module. The *Validation* and *Model tuning* ensure the model works optimally, verified using a validation dataset. When the model is working well, it can be *Served*, with the expectation to *Monitor model performance*, to continuously consider production challenges, such as data distribution changes, training-serving skew, or data quality issues (Oladele, 2021).

3.4.4. Layered view

The layered view depicts the allocation of software modules into various layers based on a unidirectional “allowed-to-use” link between the layers (Richards, 2015). We decided to base our layered view on the standard of the 5C layered architecture, as described in 3.3. Fig. 6 depicts the layered 5C architecture for the DT-based PdM RA. Starting at the top, the layered view consists of a configuration layer with a machine optimization module. The Configuration layer relies on the Cognition logic layer that provides visualizations and performs risk analysis. The Cognition layer uses data from the Cyber layer. The Cyber layer contains sub-modules to create a simulation model at a certain abstraction level of the physical asset. The Cyber layer depends on the Conversion layer, which provides security, data quality, and the predictive maintenance module. The connection layer provides the Physical Twin proxy, acquisition, storage, and communication.

4. Validation

We used the case studies described in Section 2.2 to evaluate the RA by designing an AA. We used the AA design approach proposed by Tummers, Kassahun, & Tekinerdogan, 2021, Tummers, Tobi, et al. (2021).

4.1. Case study 1: New Holland Tractor

da Silva et al. (2019) used a dataset from single manual data extraction from a New Holland T8 in Brazil. They used eight pressure

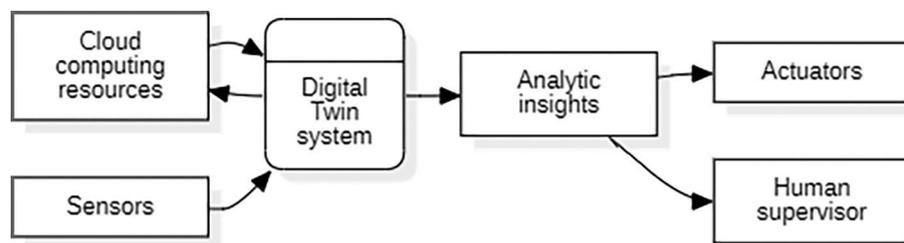


Fig. 4. Data Flow Diagram Level-0 for a DT-based PdM system RA.

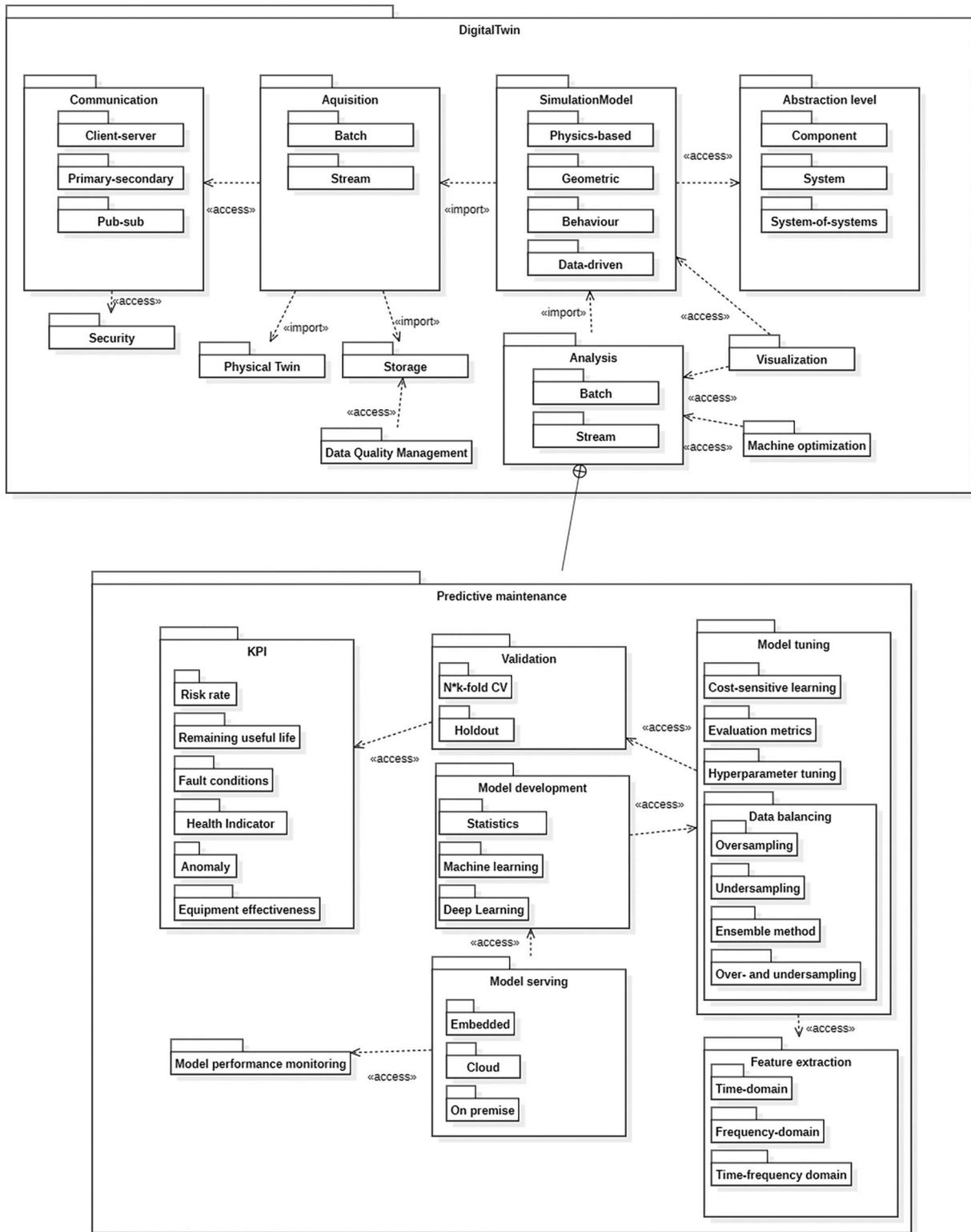


Fig. 5. Package diagram of the DT-based PdM system RA.

features from four clutches to detect anomalies. The study depicts the design of the software and tools used. We developed an AA as if the case study used a DT-based PdM system.

4.1.1. Feature diagram

We carefully investigated the study to detect the features implemented by (da Silva et al., 2019). As they used a PicoScope6 device to

perform the data acquisition and predictive maintenance, we chose Batch acquisition and analysis. The transmission system is well-defined, and as CAD models are available, the Digital Twin may be represented using physics-based techniques. Clutch analysis was based on time-series data, and anomalies were detected from the time domain. As the defects are clearly defined, traditional Machine Learning techniques for anomaly detection are sufficient. The feature diagram is shown in

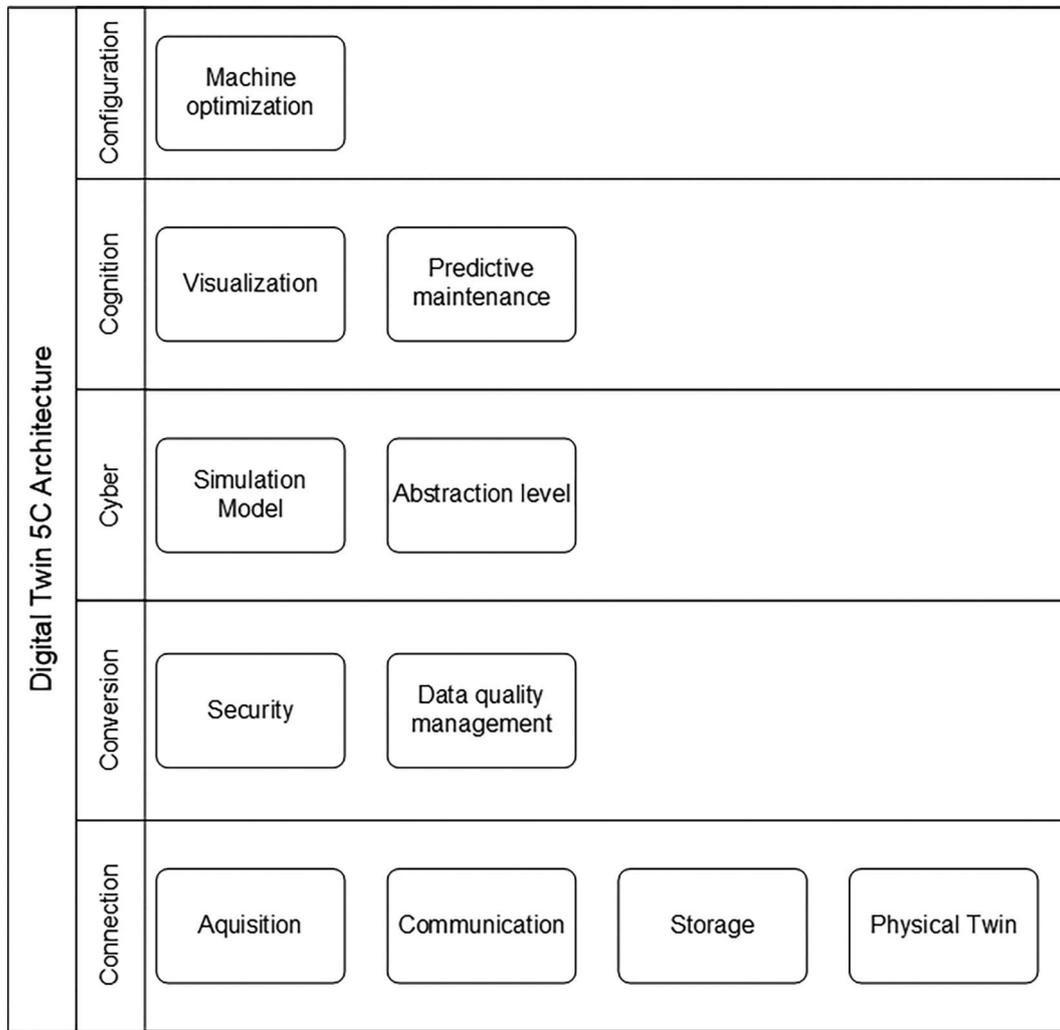


Fig. 6. Digital Twin Layered 5C Architecture.

Fig. 7.

4.1.2. User's view

We created a context diagram based on the available information explicitly mentioned in da Silva et al. (2019). There were no entities to add. We used three out of four entities from the RA. The context diagram is shown in Fig. 8.

4.1.3. Structured view

The package diagram extracted from da Silva et al. (2019) is depicted in Fig. 9. The package diagram was based on the features gathered in the section above.

4.1.4. Layered view

For the layered view, we designed a 5C architecture. The 5C diagram contains the high-level components and shows which level of the 5C architecture the components are located. These components may only communicate with components on the same or adjacent level. The 5C diagram is presented in Fig. 10.

4.1.5. Deployment view

The deployment diagram shows the two devices of the DT-based PdM system. The ClutchSystem contains components for secure communication and data acquisition of the sensors, while the ConditionMonitoringSystem contains the other components and replaces the PicoScope6 capabilities. The deployment diagram is presented in

Fig. 11.

4.2. Case study 2: semiconductor wafer manufacturing

Cohen and Singer (2021) developed a statistical process control (SPC) for identifying abnormalities, which they applied in semiconductor wafer manufacturing. More precisely, they monitored a polishing machine to optimize the quality of the wafers (i.e., achieve the optimal thickness consistently). When the polishing quality is not optimal, the machine should be maintained, which is a predictive process maintenance example.

4.2.1. Feature diagram

We carefully read the study to detect the features implemented by Cohen and Singer (2021), who proposed a smart controller framework. As such, the features were well-defined. As they used a Smart Gateway module for the Industrial Internet of Things, we assumed they used a publish-subscribe architecture for communication. Furthermore, Cohen and Singer (2021) described that the SPC is a real-time system. As such, we selected Stream for data acquisition and analysis. For root-cause analysis, the authors used Machine Learning algorithms. Specifically, they used an ordinal decision tree algorithm with a depth of 3. The feature diagram is shown in Fig. 12.

4.2.2. User's view

We created a context diagram based on the proposed SPC framework

and its main elements. We replaced Cloud Computing recourses with *Industrial Internet of Things* and *Internet/Web*. We also added *Actuators*. However, actuators are not typically included in the Digital Monitor pattern (Tekinerdogan & Verdouw, 2020). The context diagram is shown in Fig. 13.

4.2.3. Structured view

The package diagram extracted from Cohen and Singer (2021) is depicted in Fig. 14. The package diagram was based on the features from the feature diagram.

4.2.4. Layered view

For the layered view, we designed a 5C architecture. Compared to the 5C diagram of the New Holland Tractor case, this diagram also includes a *Machine Optimization* module on the configuration level. The 5C diagram is presented in Fig. 15.

4.2.5. Deployment view

The deployment diagram shows the two devices of the DT-based PdM system. The *WaferPolishingSystem* contains components for secure communication and data acquisition of the sensors, while the *Smart-ControllerServer* contains the other components and replaces the SPC capabilities. The deployment diagram is presented in Fig. 16.

4.3. Case study 3: electro-optical imaging system

We carefully investigated the study to detect the features implemented by Yu et al. (2021), who proposed a DT-based health monitoring system for an electro-optical imaging system. Yu et al. (2021) used image data to determine the health state of the electro-optical device. The image quality deteriorates over time and is the best source to monitor.

4.3.1. Feature diagram

Yu et al. (2021) clearly defined the software architecture of the Digital Twin, which is on a system level. (Yu et al., 2021) described that they used a hybrid Digital Twin model based on physical models and Bayesian, data-driven modeling. Furthermore, Yu et al. (2021) did not specify which communication protocol was used: “*The physical system requires standard data communication devices to achieve uniform conversion of different communication interfaces or protocols and uniform packaging of data.*” They extracted features from the time-frequency domain using Dirichlet Process Mixture Models. At last, Yu et al. (2021) used a statistical Gaussian Particle Filter model to determine the health indicator of the electro-optical system, which was evaluated using the Mean Squared Error metric. The feature diagram is shown in Fig. 17.

4.3.2. User's view

We created a context diagram based on the available information explicitly mentioned in Yu et al. (2021). There were no entities to add. We leveraged all entities from the RA. The context diagram is shown in Fig. 18.

4.3.3. Structured view

The package diagram extracted from Yu et al. (2021) is depicted in Fig. 19. The package diagram was based on the features from the feature diagram.

4.3.4. Layered view

For the layered view, we designed a 5C architecture. The 5C diagram for the New Holland Tractor case has the same structure as the 5C architecture for the electro-optical system. The 5C diagram is presented in Fig. 20.

4.3.5. Deployment view

The deployment diagram shows the two devices of the DT-based PdM

system. The *ElectroOpticalDevice* contains components for secure communication, data acquisition of the sensors, and data quality management. (Yu et al., 2021) describes: “*data is standardized, cleaned and packaged by physical system, and then uploaded to the Digital Twin model in digital world.*” The *Cloud* contains the other components and replaces the health monitoring system capabilities. The deployment diagram is presented in Fig. 21.

5. Discussion

5.1. General discussion

To our knowledge, this is the first RA exclusively designed for DT-based PdM systems using standardized software architecture methodologies. In this discussion, we critically reflect on our results regarding the research questions.

RQ1: What are the existing reference architectures for Digital Twin-based predictive maintenance systems?

As described in the Introduction section and Section 3.1 Domain analysis, several existing Digital Twin-based predictive maintenance systems exist. (Centomo et al., 2020; Cohen & Singer, 2021; Lee et al., 2015; Z. Liu, Jin, et al., 2018; Z. Liu, Zhang, et al., 2018; Luo et al., 2020; Mihai et al., 2021; Nordal & El-Thalji, 2021; Xiong et al., 2021) are the articles from the Domain Analysis that present RAs and frameworks.

RQ2: What is a feasible Reference Architecture for a Digital Twin-based predictive maintenance system?

We used the domain analysis information from our systematic literature review (van Dinter et al., 2022). A more comprehensive data collection with other search queries and grey literature might have introduced additional insights. However, peer-reviewed scientific articles provided a solid base for designing an RA. The results from our case studies confirm this. The feature diagram included the key components of a DT-based PdM system.

Furthermore, relevant features could be added regarding the specific needs of an application. This allows the feature model to adapt to the rapidly evolving domain of DT-based PdM systems. We identified four key entities for the context diagram based on the domain analysis. These entities were used in most of the reviewed studies. The Semiconductor Manufacturing case also showed the flexibility of the design, as we added an *Actuator* entity and replaced the *Cloud Computing resources* entity for *Industrial Internet of Things* and *Internet/Web*.

RQ3: Does the Reference Architecture allow for the derivation of a specific Application Architecture?

The case studies for the derivation of AAs were based on three peer-reviewed articles (Cohen & Singer, 2021; da Silva et al., 2019; Yu et al., 2021). Although they did not always explicitly name entities, features, or deployment environments, the studies contained sufficient detail to derive the diagrams and views. The use of five views to derive the AA was demonstrated. In theory, the same procedure can generate other potential perspectives (e.g., sequence diagrams, use-case diagrams). These diagrams may be designed using specific application requirements.

The RA package diagram showed its flexibility, as defining an AA feature diagram enabled an easy adaptation of the AA package diagram. The 5C architecture showed to be robust and informative of internal communication flows. However, using high-level modules caused the 5C architecture to be rather generic—the deployment views for the three AAs separate concerns for the device and its Digital Twin. However, regarding privacy concerns of high-tech enterprises, it may be considered to embed most software and its intelligence into the machine to limit the data stream that could be intercepted. In that case, it could be considered only to place a visualization module elsewhere or keep everything on the machine.

5.2. Threats to validity

Construct validity: The construct validity of the RA design determines if it accurately measures what it claims to.

The first threat is that we assumed performing Domain Analysis and Modeling using a semi-automated systematic review (van Dinter et al., 2022) would provide a complete view. As Digital Twins and predictive maintenance on their own are rapidly growing fields, each of these domains may be further researched to combine RAs.

The second threat is the selection of views. We have chosen to provide five views for this RA using (Alhir, 1998). However, the selection of which UML model to provide for each view may contain flaws.

Internal validity: Internal validity reveals biased relationships between results, leading to structural flaws. We carefully formulated each research question to determine the required approaches, methods, and techniques to design an RA for Digital Twin-based predictive maintenance systems to overcome these biases. As there are many options to design an RA for Digital Twin-based predictive maintenance systems, and the field is relatively new, we might have missed essential research questions or made assumptions about the UML views, which may be incomplete.

External validity: The external validity shows how well the RA study's outcome can be applied to other settings. We applied triangulation to apply our RA to three case studies from varying domains with moving and static components. Likely, novel Digital Twin RAs have not yet been applied with predictive maintenance modules and vice versa. These studies have not been published in the scope of Digital Twin-based predictive maintenance systems, so they have not been included in this study regardless of their potential.

Conclusion validity: The conclusion validity measures the reproducibility of the RA design. Our study followed the RA design methodology from Tummers, Kassahun, & Tekinerdogan, 2021, Tummers, Tobi, et al. (2021). Additionally, data for the Domain Analysis was gathered from van Dinter et al. (2022), and UML standards were applied for the RA design. We deduced all conclusions from the retrieved and synthesized data based on the models to prevent subjective interpretation of the results.

6. Conclusion and future work

We executed this study as there was a lack of a Reference

Architecture (RA) for Digital Twin-based predictive maintenance (DT-based PdM) systems based on well-established software architecture principles. In this study, we aimed to solve the challenging task of developing software architectures for predictive maintenance tasks in various application domains using the knowledge gained from a literature review.

This study demonstrated that software architecture methods could be applied to DT-based PdM systems. We proposed a high-level RA for DT-based PdM systems, which could be applied regardless of its application domain. As a result, we applied the RA to design three case study Application Architectures (AA) for New Holland Tractor clutch system monitoring, Semiconductor manufacturing polishing system anomaly detection, and Electro-optical system health monitoring. As such, this is the first study developing a domain-independent RA for DT-based PdM systems based on famous software architecture principles. A robust AA can be produced using this RA within a smaller timeframe, reducing the time-to-market.

Our future work will focus on applying our RA for DT-based PdM systems on use cases in several high-tech domains. We will mainly focus on the maintenance of the Dutch Mid-Voltage electricity network, and the maintenance of assets for semiconductor manufacturing. The insights gathered from the implementation of the framework will lead to further improvements due to iterative design changes. Furthermore, the next step of our future work will focus on developing design patterns for digital twins-based predictive maintenance systems. These can regard available data, the challenge to solve, or the type of asset or maintenance. These two studies should support industry professionals with faster time to market for developing robust predictive maintenance systems.

CRedit authorship contribution statement

Raymon van Dinter: Conceptualization, Methodology, Writing – review & editing. **Bedir Tekinerdogan:** Conceptualization, Methodology, Writing – review & editing, Supervision. **Cagatay Catal:** Conceptualization, Methodology, Writing – review & editing, Supervision.

Data availability

No data was used for the research described in the article.

Appendix A. Appendix

A.1. Application architecture New Holland Tractor case

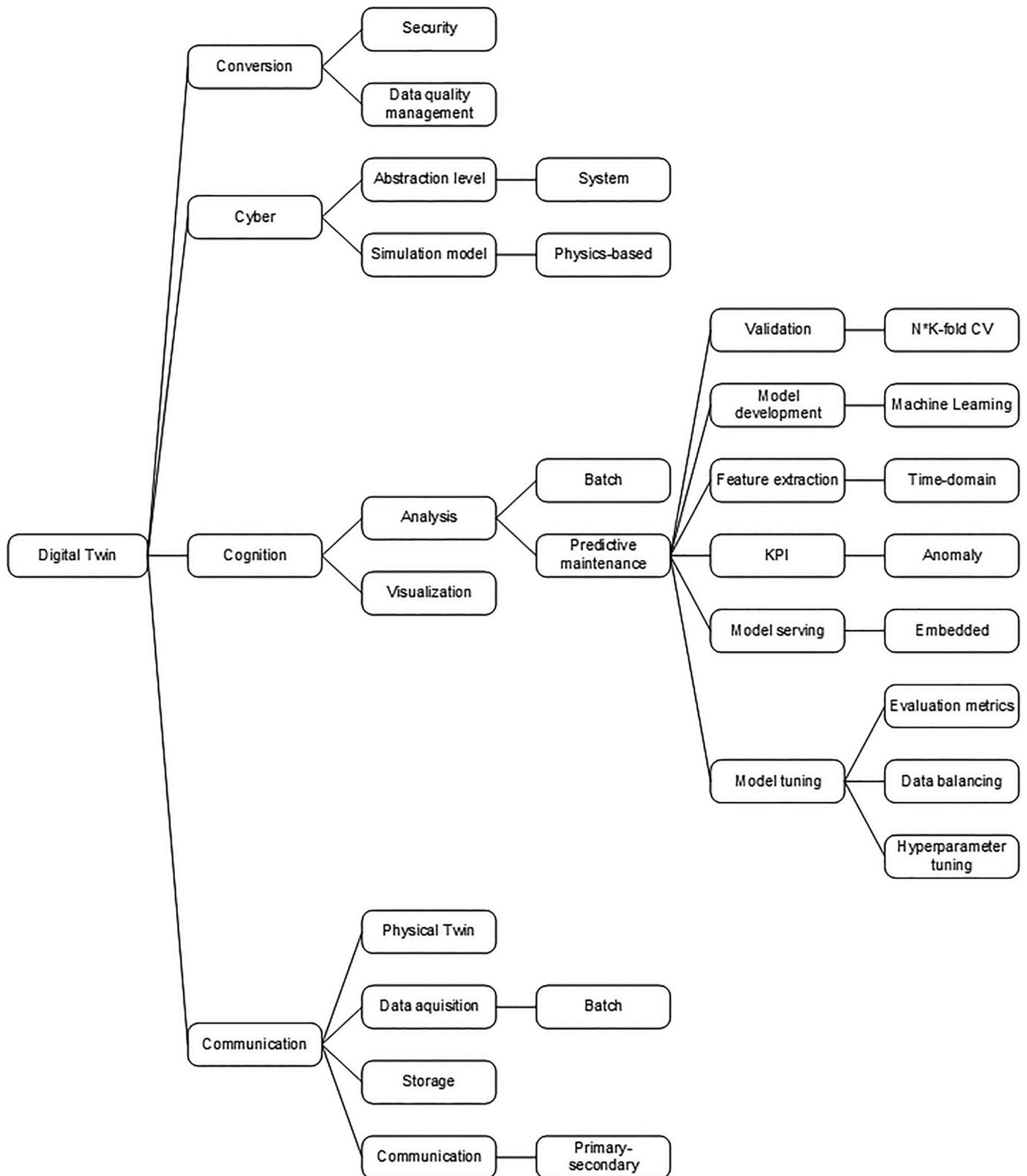


Fig. 7. Feature diagram for the New Holland Tractor Digital Twin-based predictive maintenance system based on da Silva et al. (2019).

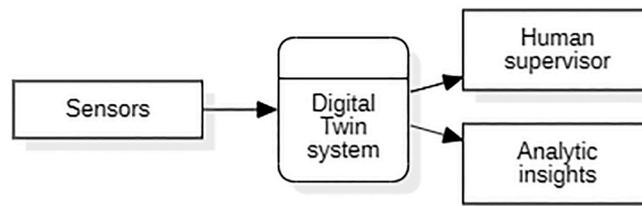


Fig. 8. Context diagram for the New Holland Tractor Digital Twin-based predictive maintenance system based on da Silva et al. (2019).

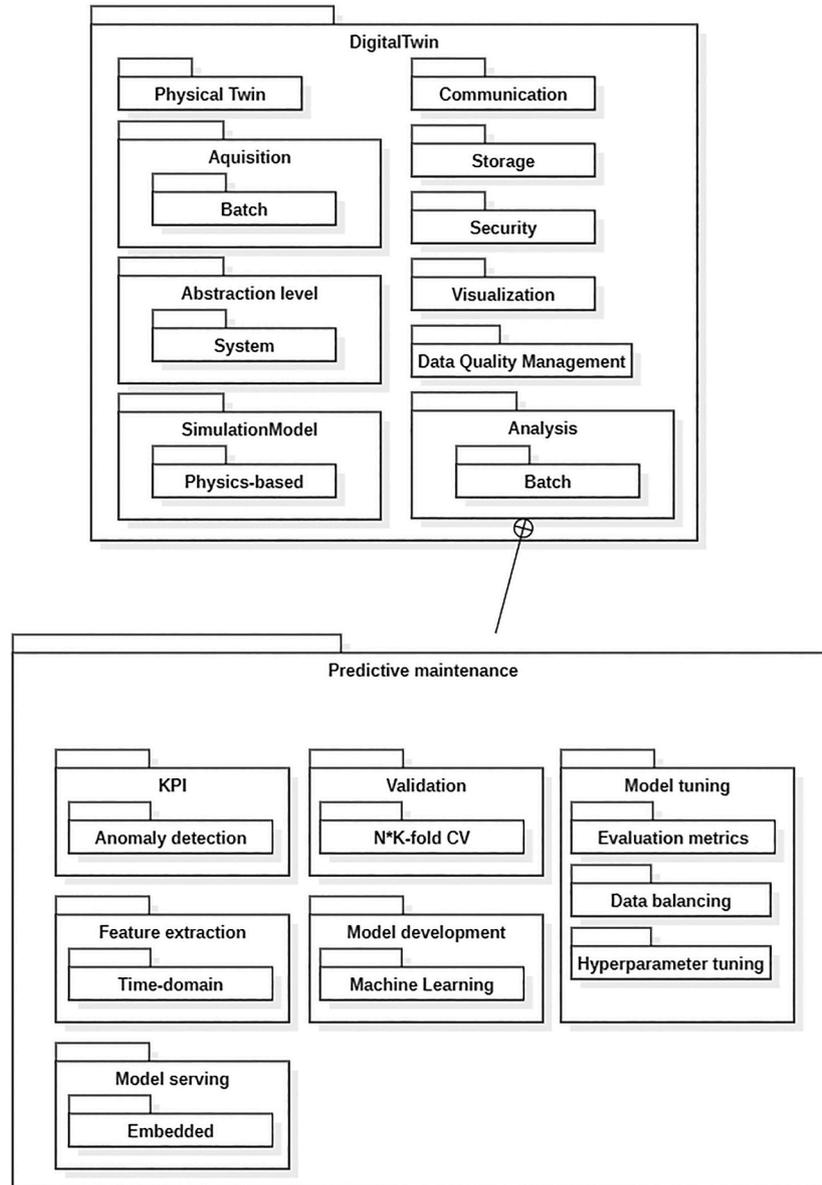


Fig. 9. Package diagram for the New Holland Tractor Digital Twin-based predictive maintenance system based on da Silva et al. (2019).

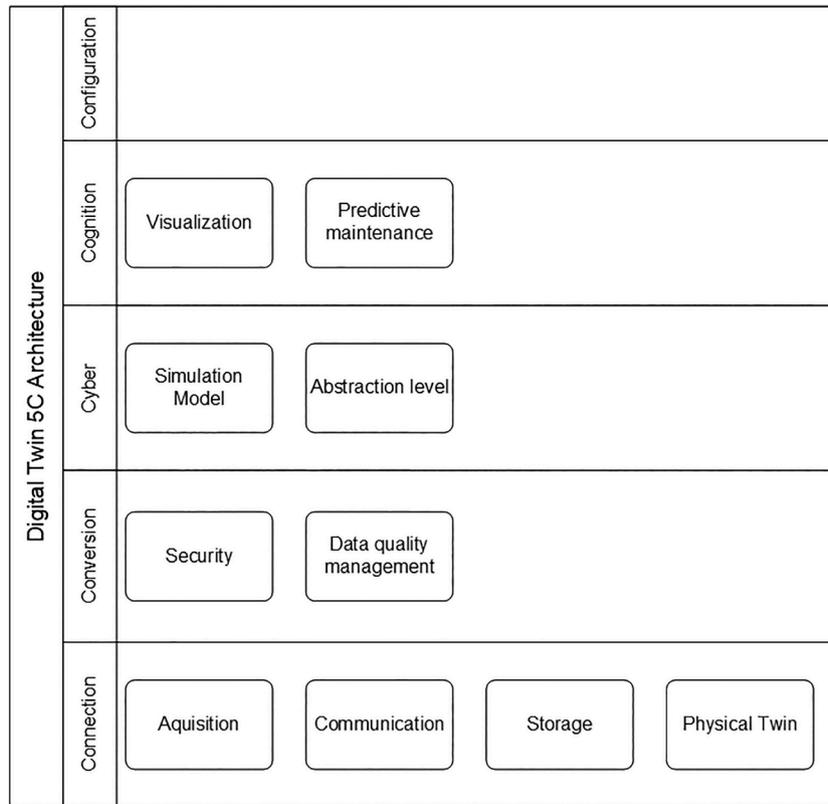


Fig. 10. 5C architecture for the New Holland Tractor Digital Twin-based predictive maintenance system based on da Silva et al. (2019).

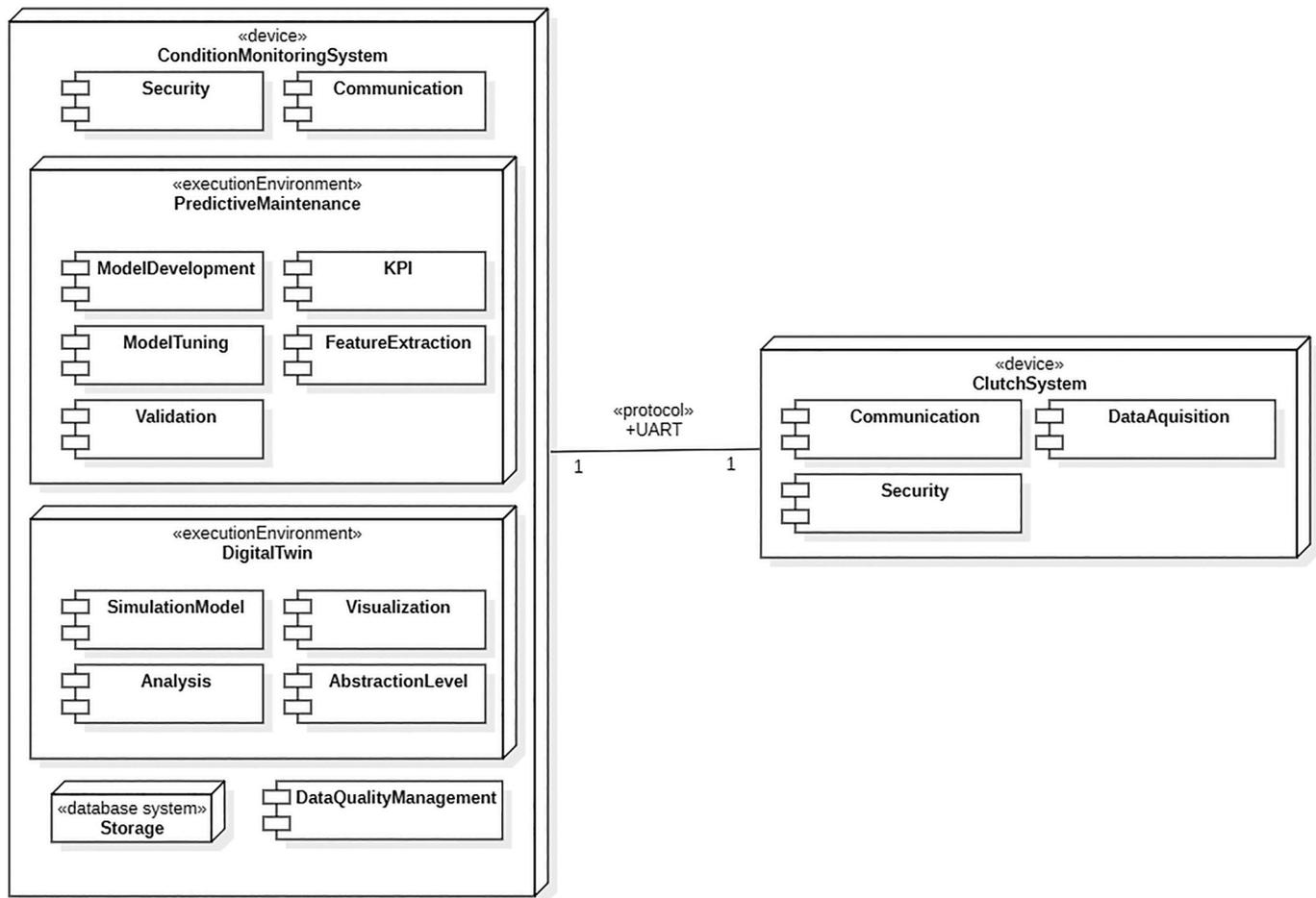


Fig. 11. Deployment diagram for the New Holland Tractor Digital Twin-based predictive maintenance system based on da Silva et al. (2019).

A.2. Application architecture semiconductor wafer manufacturing case

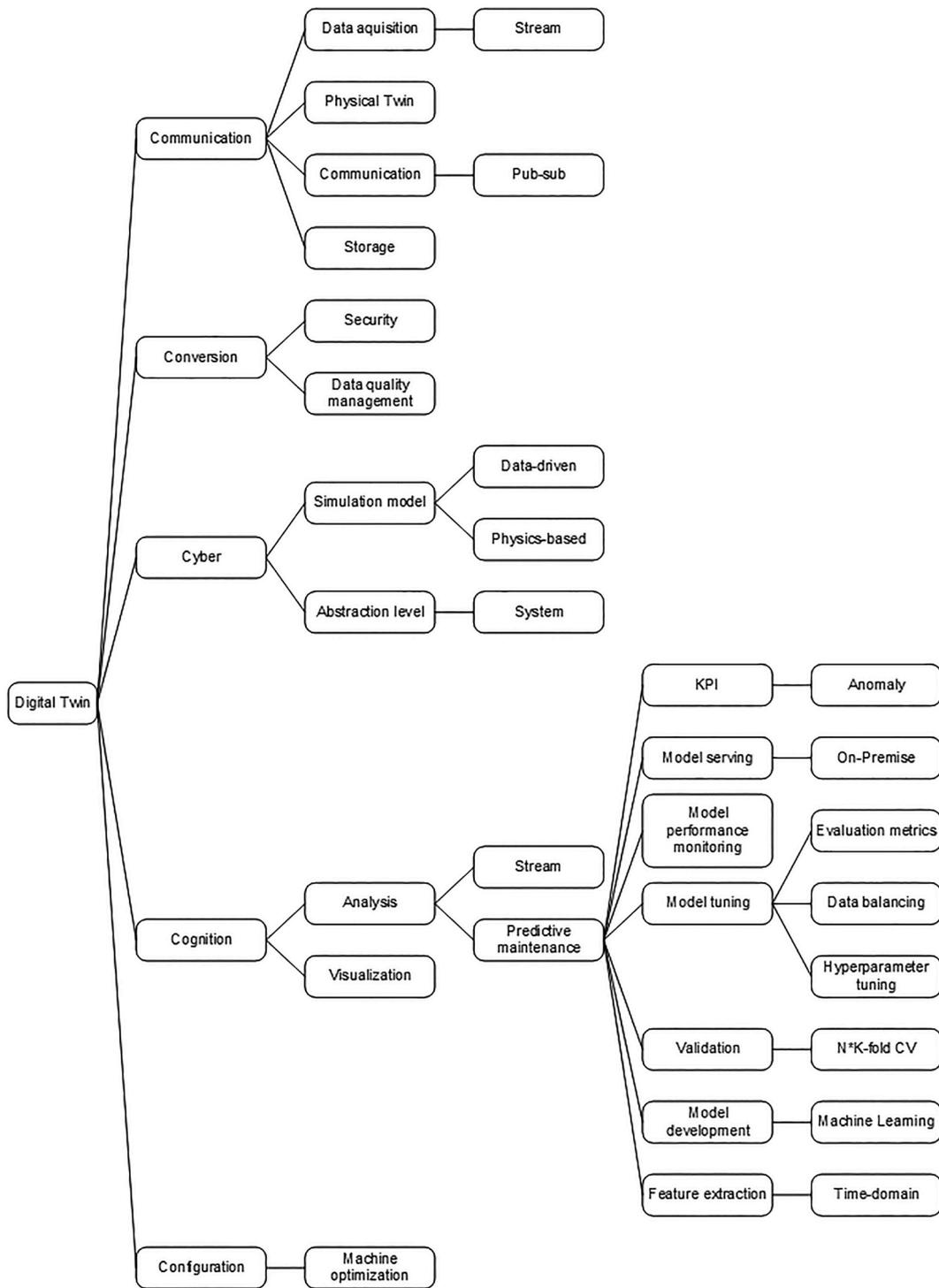


Fig. 12. Feature diagram for the semiconductor wafer manufacturing Digital Twin-based predictive maintenance system based on Cohen and Singer (2021).

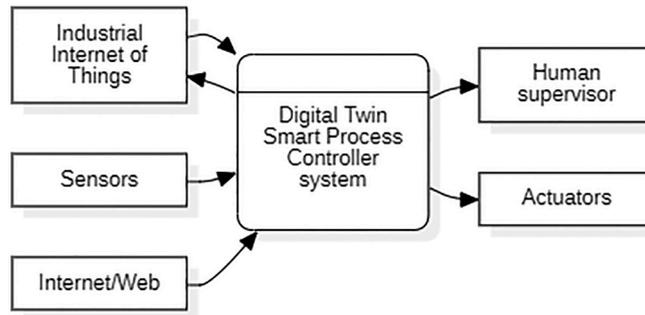


Fig. 13. Context diagram for the semiconductor wafer manufacturing Digital Twin-based predictive maintenance system based on Cohen and Singer (2021).

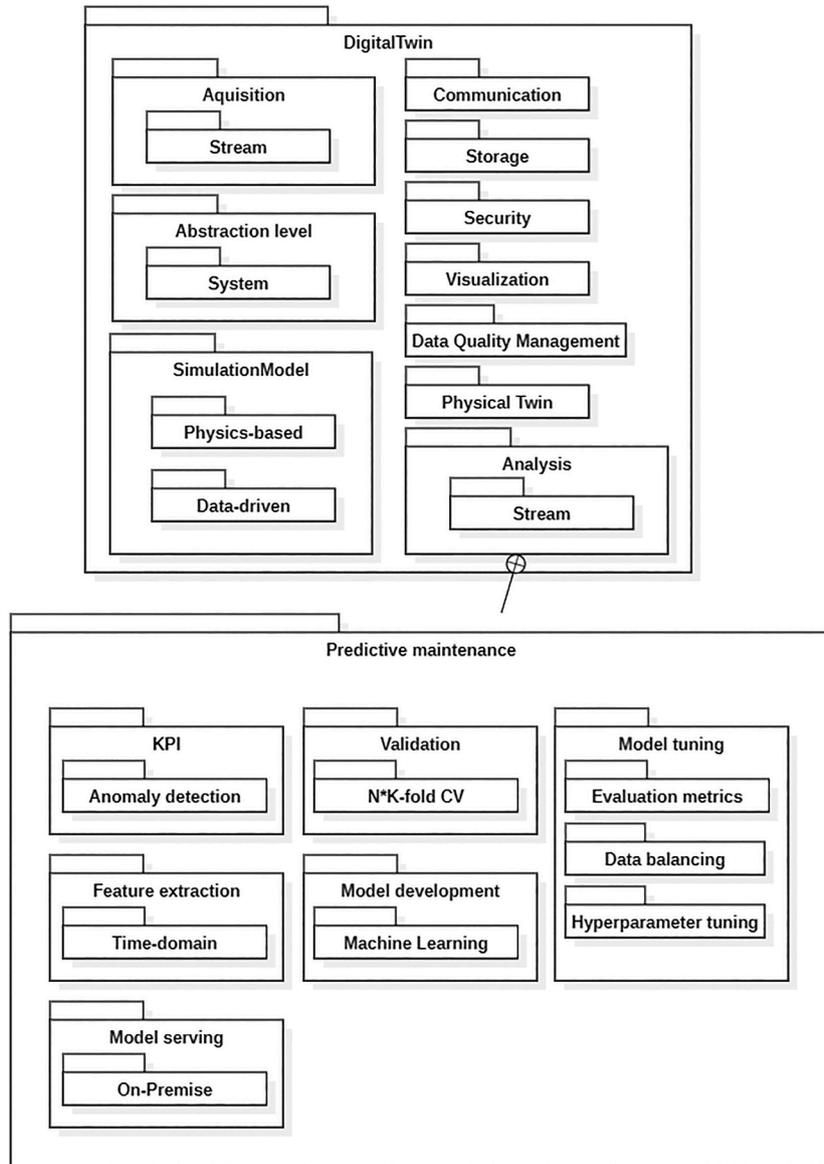


Fig. 14. Package diagram for the semiconductor wafer manufacturing Digital Twin-based predictive maintenance system based on Cohen and Singer (2021).

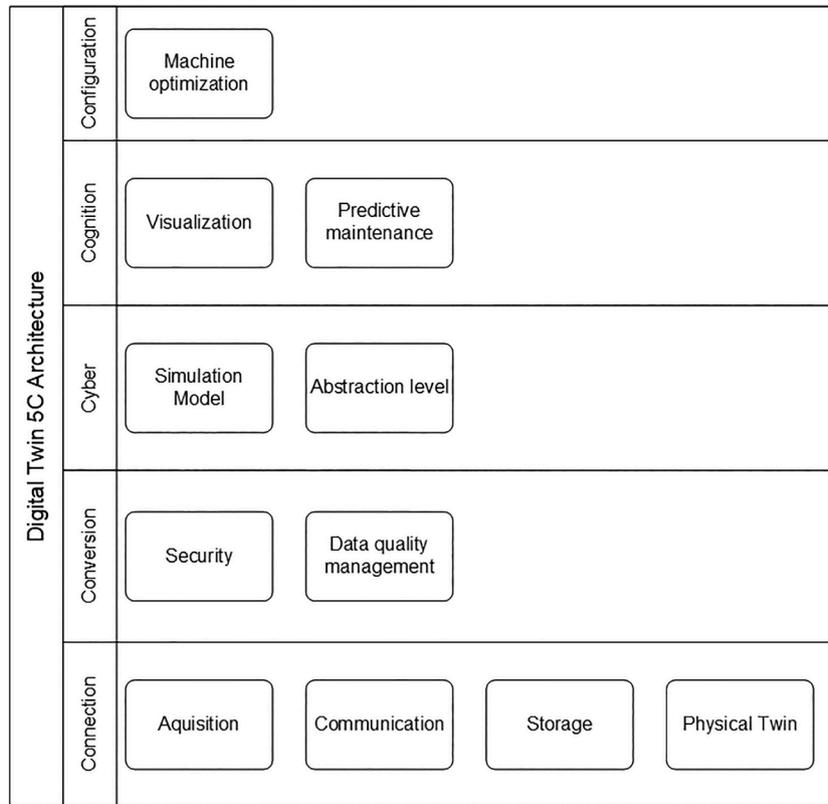


Fig. 15. 5C architecture for the semiconductor wafer manufacturing Digital Twin-based predictive maintenance system based on Cohen and Singer (2021).

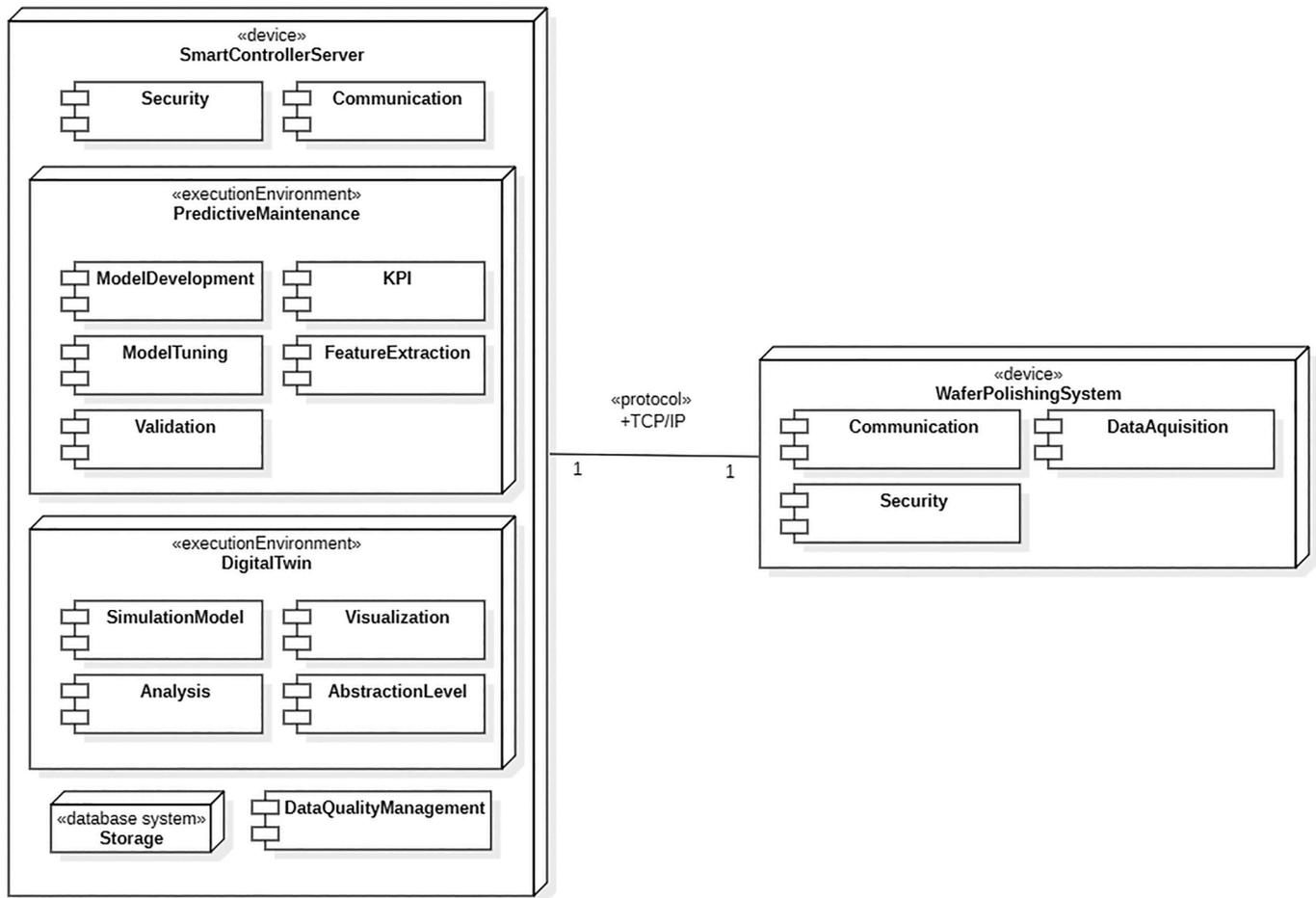


Fig. 16. Deployment diagram for the semiconductor wafer manufacturing Digital Twin-based predictive maintenance system based on Cohen and Singer (2021).

A.3. Application architecture electro-optical imaging system

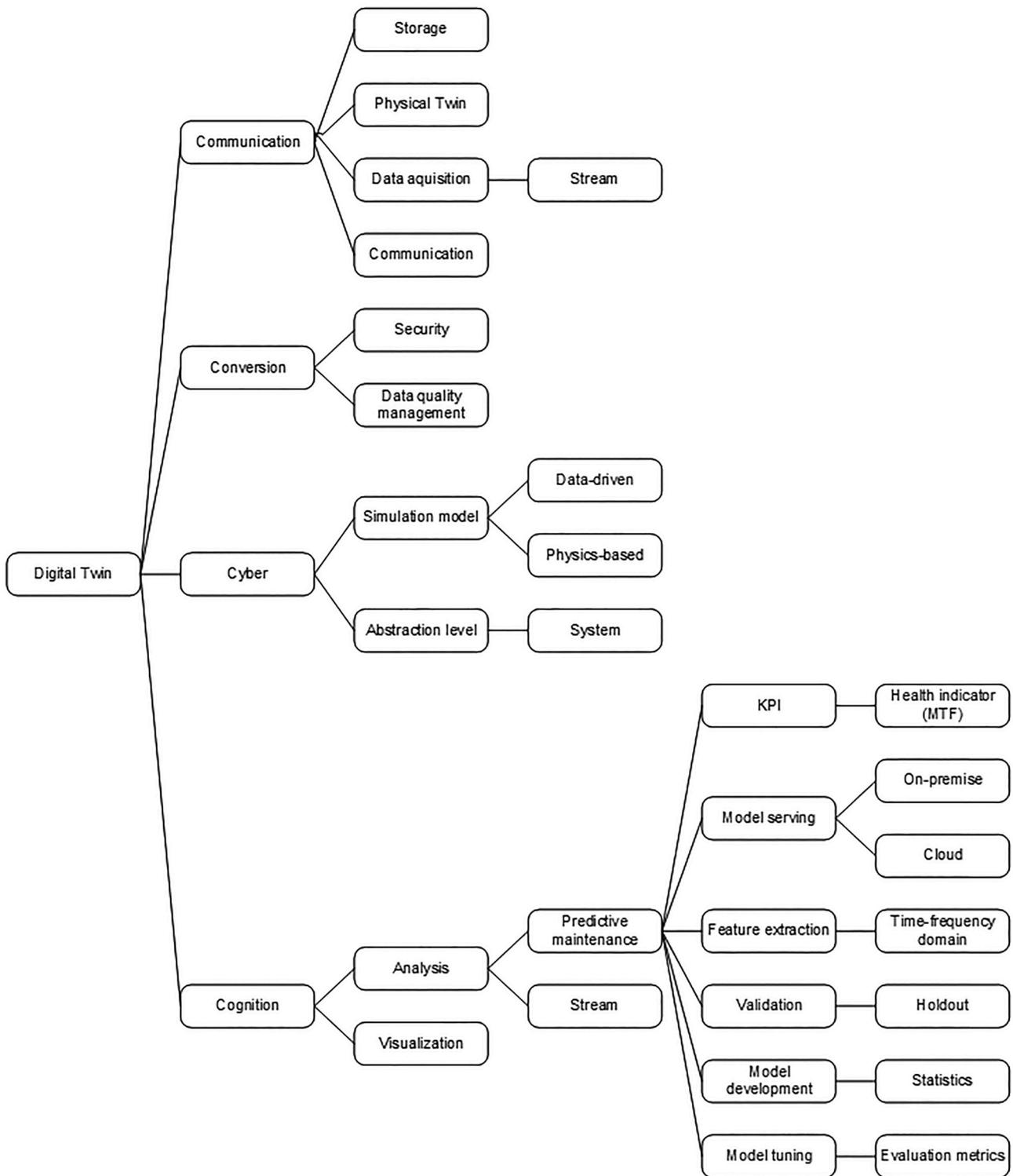


Fig. 17. Feature diagram for the electro-optical imaging Digital Twin-based predictive maintenance system based on Yu et al. (2021).

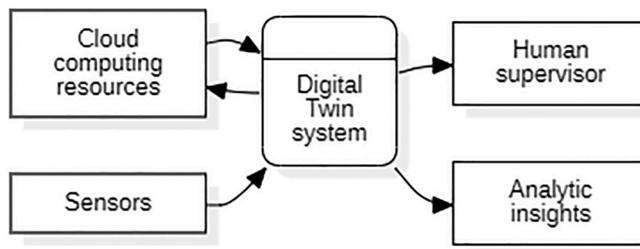


Fig. 18. Context diagram for the electro-optical imaging Digital Twin-based predictive maintenance system based on Yu et al. (2021).

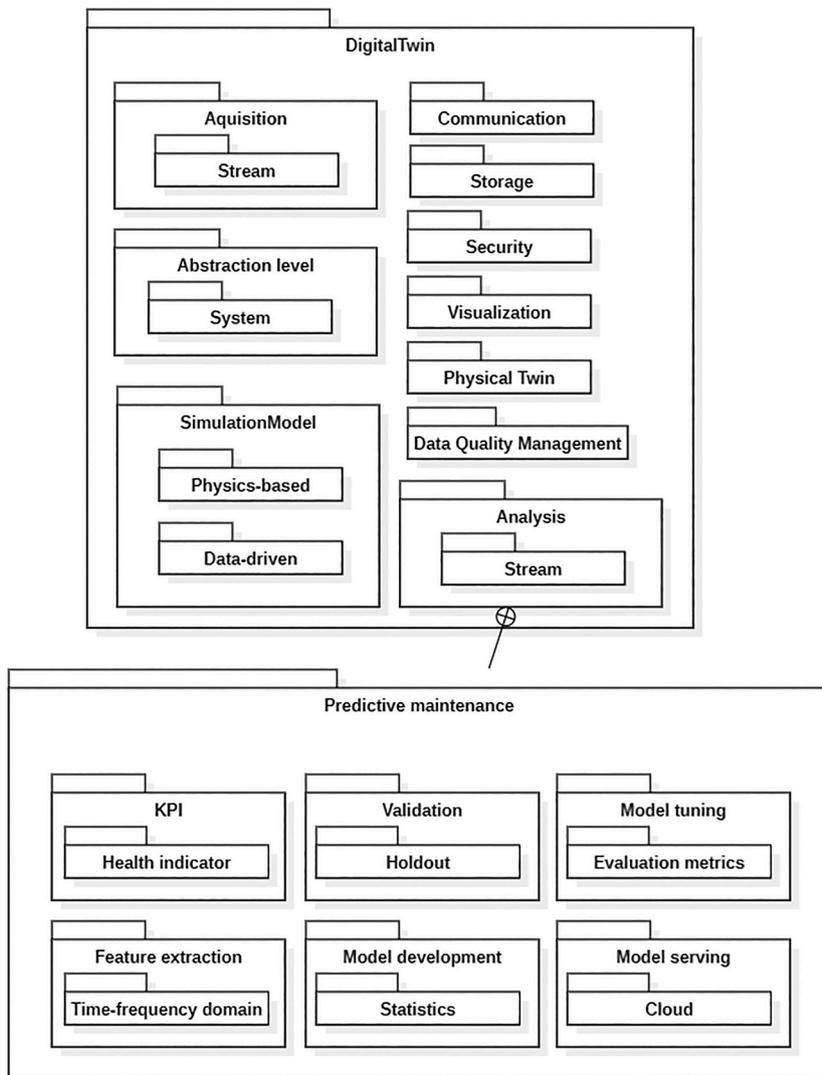


Fig. 19. Package diagram for the electro-optical imaging Digital Twin-based predictive maintenance system based on Yu et al. (2021).

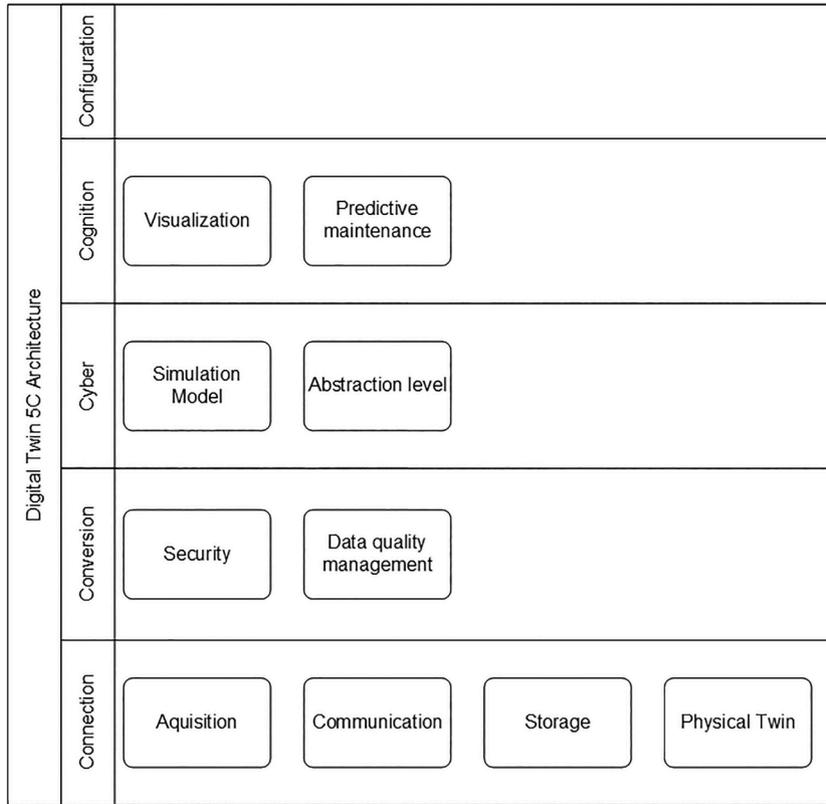


Fig. 20. 5C architecture for the electro-optical imaging Digital Twin-based predictive maintenance system based on Yu et al. (2021).

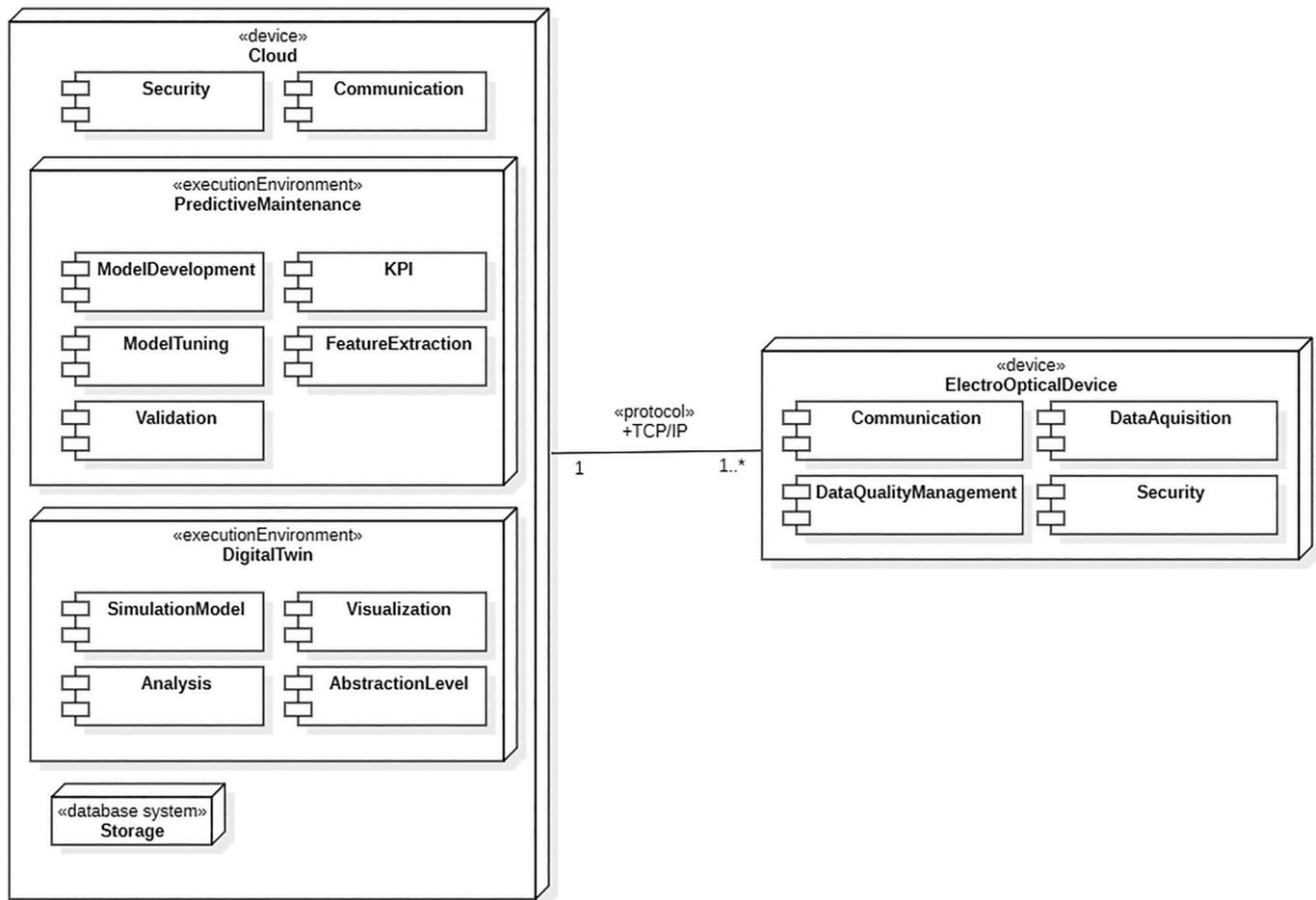


Fig. 21. Deployment diagram for the electro-optical imaging Digital Twin-based predictive maintenance system based on Yu et al. (2021).

References

- Afyenni, R. (2014). Perancangan data flow diagram untuk Sistem informasi sekolah (studi kasus pada sma pembangunan Laboratorium unp). *Jurnal Teknoif Teknik Informatika Institut Teknologi Padang*, 2(1), 35–39.
- Aheleroff, S., Xu, X., Zhong, R. Y., & Lu, Y. (2021). Digital twin as a service (DTaaS) in industry 4.0: An architecture reference model. *Advanced Engineering Informatics*, 47, Article 101225.
- Aivaliotis, P., Arkouli, Z., Georgoulas, K., & Makris, S. (2021). Degradation curves integration in physics-based models: Towards the predictive maintenance of industrial robots. *Robotics and Computer-Integrated Manufacturing*, 71.
- Aivaliotis, P., Georgoulas, K., Arkouli, Z., & Makris, S. (2019). Methodology for enabling Digital Twin using advanced physics-based modelling in predictive maintenance. *Procedia CIRP*, 81, 417–422.
- Aivaliotis, P., Georgoulas, K., & Chrysolouris, G. (2019). The use of Digital Twin for predictive maintenance in manufacturing. *International Journal of Computer Integrated Manufacturing*, 32(11), 1067–1080.
- Aivaliotis, P., Xanthakis, E., & Sardelis, A. (2020). Machines' behaviour prediction tool (BPT) for maintenance applications. In (3 ed., vol. 53. *IFAC-PapersOnLine* (pp. 325–329).
- Alhir, S. S. (1998). *UML in a nutshell: A desktop quick reference*. O'Reilly Media, Inc.
- Altun, C., & Tavli, B. (2019). Social internet of Digital Twins via distributed ledger technologies: Application of predictive maintenance. In *2019 27th telecommunications forum (TELFOR)* (pp. 1–4).
- Anis, M. D., Taghipour, S., & Lee, C. G. (2020). Optimal RUL estimation: A state-of-art Digital Twin application. In *2020 annual reliability and maintainability symposium (RAMS)* (pp. 1–7).
- Balogh, Z., Gatial, E., Barbosa, J., Leitão, P., & Matejka, T. (2018). Reference architecture for a collaborative predictive platform for smart maintenance in manufacturing. In *In: 2018 IEEE 22nd international conference on intelligent engineering systems (ines)* (pp. 000299–000304). IEEE.
- Barkalov, S., Dorofeev, D., Fedorova, I., & Polovinkina, A. (2021). Application of digital twins in the management of socio-economic systems. In , vol. 244. *E3S web of conferences*.
- Booyse, W., Wilke, D. N., & Heyns, S. (2020). Deep digital twins for detection, diagnostics and prognostics. *Mechanical Systems and Signal Processing*, 140.
- Cattaneo, L., & MacChi, M. (2019). A digital twin proof of concept to support machine prognostics with low availability of run-to-failure data. In (10 ed., vol. 52. *IFAC-PapersOnLine* (pp. 37–42).
- Centomo, S., Dall'ora, N., & Fummi, F. (2020). The design of a digital-twin for predictive maintenance. In , vol. 2020-September. *IEEE symposium on emerging technologies and factory automation, ETFA* (pp. 1781–1788).
- Deebak, B. D., & Al-Turjman, F. (2022). Digital-twin assisted: Fault diagnosis using deep transfer learning for machining tool condition. *International Journal of Intelligent Systems*, 37, 10289–10316.
- Desai, P. S., Granja, V., Higgs, C. F., & III. (2021). Lifetime prediction using a tribology-aware, deep learning-based digital twin of ball bearing-like tribosystems in oil and gas. *Processes*, 9(6).
- Dhada, M., Hernández, M. P., Palau, A. S., & Parlikad, A. K. (2021). Comparison of agent deployment strategies for collaborative prognosis. In *2021 IEEE international conference on prognostics and health management (ICPHM)* (pp. 1–8).
- Digital Twin Consortium. (2023). *Introducing an innovative framework for delivering digital twin projects based on use case capabilities* (vol. 2022).
- van Dinter, R., Tekinerdogan, B., & Catal, C. (2022). Predictive maintenance using digital twins: A systematic literature review. *Information and Software Technology*, 107008.
- Errandonea, I., Beltrán, S., & Arrizabalaga, S. (2020). Digital twin for maintenance: A literature review. *Computers in Industry*, 123, Article 103316.
- van Geest, M., Tekinerdogan, B., & Catal, C. (2021). Design of a reference architecture for developing smart warehouses in industry 4.0. *Computers in Industry*, 124, Article 103343.
- Giray, G., & Catal, C. (2021). Design of a data management reference architecture for sustainable agriculture. *Sustainability*, 13(13), 7309.
- Groba, C., Cech, S., Rosenthal, F., & Gossling, A. (2007). Architecture of a predictive maintenance framework. In *In: 6th international conference on computer information systems and industrial management applications (CISIM'07)* (pp. 59–64). IEEE.
- Heim, S., Clemens, J., Steck, J. E., Basic, C., Timmons, D., & Zwiener, K. (2020). Predictive maintenance on aircraft and applications with digital twin. In *2020 IEEE international conference on big data (big data)* (pp. 4122–4127).

- Ingeno, J. (2018). *Software Architect's handbook: Become a successful software architect by implementing effective architecture concepts*. Packt Publishing Ltd.
- Johansen, S. S., & Nejad, A. R. (2019). On digital twin condition monitoring approach for drivetrains in marine applications. In , vol. 10. *Proceedings of the international conference on offshore mechanics and arctic engineering - OMAE*.
- Josifovska, K., Yigitbas, E., & Engels, G. (2019). Reference framework for digital twins within cyber-physical systems. In 2019 *IEEE/ACM 5th international workshop on software engineering for smart cyber-physical systems (SEsCPS)* (pp. 25–31). IEEE.
- Kaul, T., Bender, A., & Sextro, W. (2020). Digital twin for reliability analysis during design and operation of mechatronic systems. In *Proceedings of the 29th European safety and reliability conference, ESREL 2019* (pp. 2340–2347).
- Kruchten, P. B. (1995). The 4+1 view model of architecture. *IEEE Software*, 12(6), 42–50.
- Lee, J., Bagheri, B., & Kao, H.-A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*, 3, 18–23.
- Lehner, D., Wolny, S., Mazak-Huemer, A., & Wimmer, M. (2020). Towards a reference architecture for leveraging model repositories for digital twins. In , Vol. 1. *2020 25th IEEE international conference on emerging technologies and factory automation (ETFA)* (pp. 1077–1080). IEEE.
- Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *The Journal of Machine Learning Research*, 18(1), 559–563.
- Liu, C., Mauricio, A., Qi, J., Peng, D., & Gryllias, K. (2020). Domain adaptation digital twin for rolling element bearing prognostics. In (1 ed.), vol. 12. *Proceedings of the annual conference of the prognostics and health management society, PHM*.
- Liu, Z., Jin, C., Jin, W., Lee, J., Zhang, Z., Peng, C., & Xu, G. (2018). Industrial AI enabled prognostics for high-speed railway systems. In 2018 *IEEE international conference on prognostics and health management (ICPHM)* (pp. 1–8).
- Liu, Z., Zhang, Z., Xu, G., Jin, W., & Lee, J. (2018). Design of cyber-physical systems architecture for prognostics and health management of high-speed railway transportation systems. *International Journal of Prognostics and Health Management*, 9(3).
- Luo, W., Hu, T., Ye, Y., Zhang, C., & Wei, Y. (2020). A hybrid predictive maintenance approach for CNC machine tool driven by digital twin. *Robotics and Computer-Integrated Manufacturing*, 65.
- Mall, R. (2018). *Fundamentals of software engineering*. PHI Learning Pvt. Ltd.
- Meraghni, S., Terrissa, L. S., Yue, M., Ma, J., Jemei, S., & Zerhouni, N. (2021). A data-driven digital-twin prognostics method for proton exchange membrane fuel cell remaining useful life prediction. *International Journal of Hydrogen Energy*, 46(2), 2555–2564.
- Mihai, S., Davis, W., Hung, D. V., Trestian, R., Karamanoglu, M., Barn, B., Prasad, R., Venkataraman, H., & Nguyen, H. X. (2021). A digital twin framework for predictive maintenance in industry 4.0.
- MLOps. (2023). *An overview of the end-to-end machine learning workflow* (vol. 2022).
- Moghadam, F. K., & Nejad, A. R. (2022). Online condition monitoring of floating wind turbines drivetrain by means of digital twin. *Mechanical Systems and Signal Processing*, 162.
- Moghadam, F. K., Rebouças, G. F. S., & Nejad, A. R. (2021). Digital twin modeling for predictive maintenance of gearboxes in floating offshore wind turbine drivetrains. *Forschung im Ingenieurwesen/Engineering Research*, 85(2), 273–286.
- Moi, T., Cibicik, A., & Rølvåg, T. (2020). Digital twin based condition monitoring of a knuckle boom crane: An experimental study. *Engineering Failure Analysis*, 112.
- Nordal, H., & El-Thalji, I. (2021). Modeling a predictive maintenance management architecture to meet industry 4.0 requirements: A case study. *Systems Engineering*, 24(1), 34–50.
- Oladele, S. (2021). *A comprehensive guide on how to monitor your models in production* (vol. 2022).
- Oluwasegun, A., & Jung, J.-C. (2020). The application of machine learning for the prognostics and health management of control element drive system. *Nuclear Engineering and Technology*, 52(10), 2262–2273.
- Priyanka, E. B., Thangavel, S., Gao, X.-Z., & Sivakumar, N. S. (2021). Digital twin for oil pipeline risk estimation using prognostic and machine learning techniques. *Journal of Industrial Information Integration*, 100272.
- Qiao, Q., Wang, J., Ye, L., & Gao, R. X. (2019). Digital twin for machining tool condition prediction. *Procedia CIRP*, 81, 1388–1393.
- Rajesh, P. K., Manikandan, N., Ramshankar, C. S., Vishwanathan, T., & Sathishkumar, C. (2019). Digital twin of an automotive brake pad for predictive maintenance. *Procedia Computer Science*, 165, 18–24.
- Redelinghuys, A., Kruger, K., & Basson, A. (2019). A six-layer architecture for digital twins with aggregation. In *International workshop on service orientation in Holonic and multi-agent manufacturing* (pp. 171–182). Springer.
- Richards, M. (2015). *Software architecture patterns* (Vol. 4). 1005 Gravenstein Highway North, Sebastopol, CA: O'Reilly Media, Incorporated.
- Rossini, R., Conzon, D., Prato, G., Pastrone, C., Reis, J., & Gonçalves, G. (2020). REPLICa: A solution for next generation iot and digital twin based fault diagnosis and predictive maintenance. In , vol. 2739. *CEUR workshop proceedings* (pp. 55–62).
- Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2), 131–164.
- Semeraro, C., Lezoche, M., Panetto, H., & Dassisti, M. (2021). Digital twin paradigm: A systematic literature review. *Computers in Industry*, 130, Article 103469.
- Short, M., & Twiddle, J. (2019). An industrial digitalization platform for condition monitoring and predictive maintenance of pumping equipment. *Sensors (Switzerland)*, 19(17).
- Cohen, Y., & Singer, G. (2021). A smart process controller framework for industry 4.0 settings. *Journal of Intelligent Manufacturing*, 32, 1975–1999.
- da Silva, C. A. G., Rodrigues de Sá, J. L., & Menegatti, R. (2019). Diagnostic of failure in transmission system of agriculture tractors using predictive maintenance based software. *AgriEngineering*, 1(1), 132–144.
- Tao, F., Zhang, M., Liu, Y., & Nee, A. Y. C. (2018). Digital twin driven prognostics and health management for complex equipment. *CIRP Annals*, 67(1), 169–172.
- Tekinerdogan, B., & Verdouw, C. (2020). Systems architecture design pattern catalog for developing digital twins. *Sensors*, 20(18), 5103.
- Tummers, J., Kassahun, A., & Tekinerdogan, B. (2021). Reference architecture design for farm management information systems: A multi-case study approach. *Precision Agriculture*, 22(1), 22–50.
- Tummers, J., Tobi, H., Catal, C., & Tekinerdogan, B. (2021). Designing a reference architecture for health information systems. *BMC Medical Informatics and Decision Making*, 21(1), 1–14.
- Tygesen, U. T., Worden, K., Rogers, T., Manson, G., & Cross, E. J. (2019). State-of-the-art and future directions for predictive modelling of offshore structure dynamics using machine learning. In *Conference proceedings of the society for experimental mechanics series* (pp. 223–233).
- Tzani, N., Andriopoulos, N., Magklaras, A., Mylonas, E., Birbas, M., & Birbas, A. (2020). A hybrid cyber physical digital twin approach for smart grid fault prediction. In , vol. 1. *2020 IEEE conference on industrial cyberphysical systems (ICPS)* (pp. 393–397).
- Uhlmann, E., Polte, J., & Koutrakis, N.-S. (2021). Holistic concept towards a reference architecture model for predictive maintenance. *Procedia CIRP*, 104, 1430–1433.
- Wang, K.-J., Lee, Y.-H., & Angelica, S. (2020). Digital twin design for real-time monitoring – A case study of die cutting machine. *International Journal of Production Research*, 1–15.
- Wang, T., Liu, Z., Liao, M., & Mrad, N. (2020). Life prediction for aircraft structure based on Bayesian inference: Towards a digital twin ecosystem. In (1 ed.), vol. 12. *Proceedings of the annual conference of the prognostics and health management society, PHM*.
- Werner, A., Zimmermann, N., & Lentjes, J. (2019). Approach for a holistic predictive maintenance strategy by incorporating a digital twin. *Procedia Manufacturing*, 39, 1743–1751.
- Xiong, M., Wang, H., Fu, Q., & Xu, Y. (2021). Digital twin-driven aero-engine intelligent predictive maintenance. *International Journal of Advanced Manufacturing Technology*, 114(11–12), 3751–3761.
- Xu, Y., Sun, Y., Liu, X., & Zheng, Y. (2019). A digital-twin-assisted fault diagnosis using deep transfer learning. *IEEE Access*, 7, 19990–19999.
- Yu, J., Song, Y., Tang, D., & Dai, J. (2021). A digital twin approach based on nonparametric Bayesian network for complex system health monitoring. *Journal of Manufacturing Systems*, 58, 293–304.
- Zenisek, J., Wolfartsberger, J., Sievi, C., & Affenzeller, M. (2018). Streaming synthetic time series for simulated condition monitoring. *IFAC-PapersOnLine*, 51(11), 643–648.
- Zhang, G., Huo, C., Zheng, L., & Li, X. (2020). An architecture based on digital twins for smart power distribution system. In *2020 3rd international conference on artificial intelligence and big data (ICAIBD)* (pp. 29–33).
- Zhidchenko, V., Startsev, E., & Handroos, H. (2022). Reference architecture for running computationally intensive physics-based digital twins of heavy equipment in a heterogeneous execution environment. *IEEE Access*.