Received: 1 September 2020 | Revised: 4 October 2020 | Accepted: 8 October 2020

DOI: 10.1002/cpe.6111

RESEARCH ARTICLE

WILEY

An effective architecture of digital twin system to support human decision making and AI-driven autonomy

Fahed Mostafa[1,2] | Longquan Tao[1,2] | Wenjin Yu[1]

[1]Computer Science and Information Technology, La Trobe University, Bundoora, Victoria, Australia

[2]Enterprise Data Services, Australia Energy Market Operator, Melbourne, Victoria, Australia

Correspondence
Longquan Tao, Computer Science and Information Technology, La Trobe University, 1 Kingsbury St, Bundoora, VIC, Australia.
Email: C.Tao@latrobe.edu.au

Abstract

With the development of IoT technologies, digital twin has become an increasingly popular concept that is considered the next generation of digitalization for decision making support (human-oriented) and even fully autonomous. However, although there are a few research projects that have proposed available digital twin architectures, they are either missing critical components or difficult to be converted into a practical application. In this article, everything we proposed had been implemented in our production environment and is facilitating our manufacturing and mining processes. It is initiated by a data analytic maturity model which formulates the evaluation route of data analytics. Then, a novel six-layer digital twin model is established that aims to set the standards. In this model, we defined that all the automated calculation jobs should be driven by digital twin metadata such as the hyperparameters of machine learning. The metadata will be updated by metadata updating feedback flow which most current digital twin projects are missing.

KEYWORDS

computational intelligence, decision making support, digital twin, hybrid system, IoT

1 | INTRODUCTION

The origin of the term "twin" in this context can be tracked back to NASA's Apollo program[1] in 1960s, where the engineers constructed two identical space vehicles to simulate conditions between the deployed craft and its twin on earth. One of the first "digital twin" concept was proposed by Främling et al. who utilized IoT technologies to simulate the lifecycle of a product item. This IoT system was named "virtual counterpart" which is considered the oldest synonym of "digital twin".[2]

The recent definition of a digital twin is a digital replication of various physical assets such as machines, people, functional areas, and the surrounding physical circumstances can be utilized to track, monitor, and intelligently predict for analytics, maintenance, and diagnostics purposes that leverages IoT technologies and able to react to the user fired or automatically triggered adjustments of its configurations. The entire suit of physical assets that the digital twin replicates is called its physical twin.

To achieve the features defined above, digital twin should integrate:

- Data collection
- Data modeling
- Domain knowledge
- Analytics
- Machine learning/artificial intelligence

Therefore, the fusion of a vast number of these different types of technologies and theories results in considerable complexity of implementing a digital twin. However, current research of the digital twin architecture with regards to its robustness, feasibility, and effectiveness are still insufficient. This article aims to establish a unified standard for such a digital twin architecture by proposing a six-layer digital twin model based on existing successful experiences in the manufacturing industry. This model has been proven its effectiveness in our scenarios, and it is also conceptually generalized so that can be applied on any digital twin projects.

During the practical digital twin building process, we have discovered that a digital twin needs some master data to guide its behaviors. For example, the parameters for cleaning ingested data and applying machine learning algorithms are critical, as they are necessary to achieve the autonomy. It also must be mutable in order to react to any changes in practice such as turnarounds of the physical equipment or changes of business rules and objectives. Therefore, we also propose a metadata updating feedback flow as a critical digital twin feature which is introduced in this article.

In the rest of the article, some related works are discussed in Section 2. In Section 3, a data analytics maturity model is proposed which formulates the regular routine of typical data analytics project development. The digital twin is usually be prototyped in the second phase as a critical milestone to start the fast-moving value creation of analytics. Section 4 proposes a novel six-layer digital twin model, as well as the metadata updating feedback flows that enable the digital twin to react to changes. In order to demonstrate the executability and effectiveness, the digital twin that has been implemented in our production environment that is assisting the manufacturing processes is shown in Section 5. Finally, the article is concluded in Section 6.

2 | RELATED WORKS

With the development of IoT and AI technologies, there is an increasing number of projects that are executed based on digital twin concepts in multiple industries predominantly in manufacturing,[3-6] mining,[7,8] and transportation.[9,10] Recently, there are also a large number of innovative researches that utilizes digital twin concepts. For example, Verner et al. developed a digital twin for a humanoid robot which can intelligently learn from warehouse workers to lift weights. Their digital twin system is responsible for not only the learning, but also monitoring the robot status and maintaining the inventory.[11] Botkina et al.[12] implemented a digital twin for cutting tools which improved deliverables quality by optimizing the cutting precision using a digital twin. Tao et al.[13] applied a digital twin to health industry through which the system is utilized to manage the complex medical equipment and achieve prognostics analysis of them.

In terms of the architecture of a typical digital twin, there are already a number of discussions available. For example, Redelinghuys et al.[14] coincidently proposed a digital twin design model with six layers which is similar to us. However, their model is excessively specialized concentrated on the connection between the physical assets and the IoT system and does not go beyond the concepts of IoT technologies.[15]

By contrast, the architecture that is proposed by Alam and El Saddik is more practical,[16] which is named CPS (cyber-physical system). The CPS consists of physical things such as machines and cyber things that are digital replications of these physical things. The CPS architecture is consisting of several layers such as a peer-to-peer relation layer is the place to associate cyber things together and an intelligent service layer includes all the algorithms and system usage monitoring that includes data consumption, visualization and system administration. This model describes a digital twin system and conceptually classified different types of activities. However, it is still difficult to be utilized as a guide on implementing an intelligent digital twin due to numerous missing critical concepts. For example, the data formats and structures from the physical assets in real world applications might not be able to be related naturally, so it is important to derive the relationships between them when they are ingested. There are also the security concerns and optimizations of data persistence is not mentioned in this model. More importantly, the automated jobs in the intelligent service layer should not be designed so statically. That is, a metadata updating feedback flow from the top layer to low-leveled layer is needed when the requirements of intelligent services are changed, and this is missing in the CPS.

In our proposed six-layer digital twin model, all the activities of digital twins are classified and explained in depth. A feedback flow which is considered the key factor for the intelligence and autonomy is also demonstrated in section 4.

3 | ANALYTICS MATURITY MODEL

As shown in Figure 1, there is a series of processes and efforts that bring a data analytic model to maturity, where more effort and time is spent on different activities shown on the curve, the business value it creates will be raised exponentially. However, these activities need to happen at a certain time within a predetermined sequence of four different phases. The digital twin will be prototyped in phase 2 and gradually established and enriched with more advanced features. In this section, this model will be discussed more in detail.

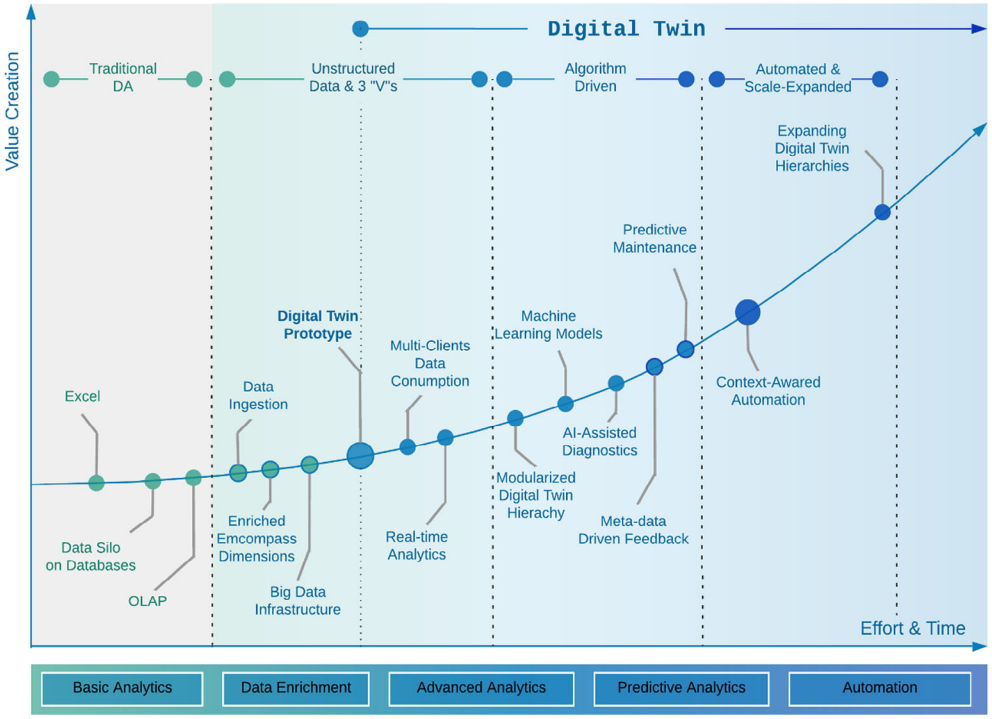


FIGURE 1 Generic data analytics maturity model

3.1 | Traditional data analysis

Most data analytic activities start with dumping data files from transactional databases and utilizing tabular tools such as Microsoft Excel to perform instant analysis, which is indeed quick and easy. However, to extend these activities onto a relatively larger dataset, a data warehouse with data silos built on SQL databases is needed, then the data views with predetermined concentrations can be obtained. A data warehouse allows users to retrieve historical data and conduct online analytical processing analysis based on one or multiple dimension tables joint with fact tables. In this case, data extracted from different data sources must be transformed into rigid structures, namely, star schema. However, due to the development of industrial control systems technologies, the sheer volume and variations of data available and captured has become overloaded.[17] Therefore, the shortcoming of data warehouses on traditional relational databases have revealed limitations when the dataset is large volume, high velocity and various (3 "V"s), for example, videos is unstructured data that are difficult to be stored in relational databases.[18]

3.2 | Involving the big data ecosystem

To solve the above-mentioned 3 "V"s problems, more capacity and capabilities are needed. These additional requirements usually urge the project to involve a big data platform, most commonly has been a data lake,[19] which is designed to solve these sophisticated issues. Once the project objective has been determined, the previously structured data from SQL databases and any other data from different sources that are relevant to the physical twin with either semistructured or unstructured data are also ingested into the centralized data platform to construct a big data infrastructure. After this, a prototype of a digital twin is established with shallow replication of the physical twin in terms of its characteristics, features and status. Therefore, multiple data consumers such as third party data analytic tools, diagnostic software and reporting tools are able to subscribe to the data feed via interfaces in the big data platform to consume the transformed data. In addition, depending on the specific use case, the data feed can be implemented as streamed based to enable real-time analytics.

(1) Encompass dimensions

In this maturity phase, it is commonly a critical requirement that the encompassed dimensions of the physical twin need to be involved to improve the effectiveness of its data analytics. The encompassed dimensions indicate those relevant data that may influence the physical twin in any aspects. For example, weather information should be considered and fed into the digital twin as parameters as it will influence the productivity of an open-pit mining plant.[20] In addition, these dimensions are usually from various different data sources, so it becomes one of the key features of digital twin. This typically requires big data techniques to transform and fuse the data of these dimensions, and therefore improve the confidence and precision of the data analytics activities.

(2) Big data ecosystem

A typical big data ecosystem is designed to overcome the issues of data with large volume, more variety and high velocity.[21] It is called an ecosystem as there might be a series of tools for various purposes that collaborate with each other. A few popular utilities of big data ecosystem are listed as examples:

- Apache Hadoop: a high availability and high I/O throughput distributed with file management system.[22]
- MapReduce: a programming framework that enables algorithms to split and shuffle the data that is distributed and process in parallel.[23]
- Apache HBase: a distributed parallelized NoSQL database with rapid random index lookup.[24]
- Apache Spark: a distributed environment-based computing framework that includes built-in machine learning algorithms.[25]

There are also a wide-range of available open-source and proprietary utilities in big data ecosystem that serve to solve the challenges faced when implementing a digital twin. Therefore, an appropriate digital twin should be designed to leverage big data techniques to satisfy these requirements.

(3) Variety of interfaces

The interfaces that a big data infrastructure that can provide data services are various due to its flexibility, in contrast to most databases are limited to ODBC/JDBC connections. For example, the RESTful API[26] is supported by most modern web applications, and the Socket API[27] is usually used for real-time communication. This allows data consumer to support real-time data analytics in the digital twin without any limitations. Therefore, a big data ecosystem-driven data analytics is considered to surpass the traditional data warehouse and data silos.

3.3 | AI integration

Next, the digital replication of the physical twin dives deeper and transits away from copying only data from its physical counterpart to also modeling its behaviors. The initial step that enables behavior modeling is to modularize the physical twin, and conceptually split a large digital twin into segmented hierarchical subdigital twins. For example, it might be necessary to create subdigital twins for specific components of a machine which has its own functionality, in order to model their behaviors atomically, as well as the collaboration between these multiple subdigital twins. Then, machine learning algorithms are involved to mathematically model these behaviors and quantify the practical problems for analysis, such as the healthy rate of a compressor. After that, the digital twin now has AI-based ability to support engineers in various activities such as diagnosing and troubleshooting the machine. Consequently, predictive maintenance can be achieved, and the digital twin is able to self-diagnose sophisticated issues before it actually happens.[28]

Take an example of a manufacturing plant for which there might be thousands of sensors for all its equipment. It turns out that not even the most experienced engineer can monitor and diagnose from such a large number of real-time inputs. However, a trained machine learning model is able to extract useful patterns from these and provide AI-assisted advice to engineers, for instance, raising alarms with severity levels. Therefore, the more accurate AI algorithms that are implemented in the digital twin system, the deeper behaviors of the physical twin are copied by the digital twin.

Machine learning algorithms should not be static in a typical digital twin system. It should be adjustable based on practical dynamic requirements that change from users and scenarios. In the same example mentioned above, the thresholds that represent the severity levels of the alarms must be adjustable if there are any changes that happen in the physical twin which are commonly happened such as machine model upgrades. Therefore, the metadata updating feedback flow was designed to enable this. Its design and implementation are introduced in the section metadata updating feedback flow in detail.

3.4 | Autonomy and scale-expansion

After all the functionalities and features are implemented in the above three phases, the digital twin arrives the final phase, the autonomous phase. By implementing more interfaces in the digital twin connected back to its physical twin, a digital twin can achieve self-diagnosing and self-repairing. For example, when a machine learning model raises an alarm, a contribution analysis can be triggered to find out which subdigital twin is responsible for this anomaly. Once the contributing factor is determined, a program that implements an interface that can control a process of the physical twin can attempt to eliminate the alarms. One possible way could be adjusting relevant controllable physical entities such as a valve or cooling system. Therefore, most of the issues that could cause serious production loss can be eliminated before they actually start to impact production. This could save considerable costs on both funds and time.

Finally, as the opposite of dividing a digital twin to subdigital twins, the boundary of a digital twin can also be expanded by involving more digital twins on the same ontology level to form a "super digital twin." For example, multiple digital twins of different functional areas in a plant can be

integrated into a super digital twin that represents a plant. The digital twins of plants can be continuously expanded into a super digital twin that replicates the whole manufacturing company, and consequently higher level of behaviors such as materials management and shipping management can be modeled in order to optimize operations at an enterprise level.

4 | DIGITAL TWIN ARCHITECTURE

In this section, a six-layer digital twin model is proposed to conceptually classify the components of a digital twin based on their responsibilities and how they should collaborate with each other. The six layers are listed as below from the top layer to the bottom layer, which is also showed in Figure 2.

1. Consumption layer
2. Service layer
3. Inference layer
4. Persistence layer
5. Ingestion layer
6. Physical layer

These layers are defined and introduced in their respective subsections. The data flows between each layer and how the Metadata Driven Feedback Flow influences the digital twin is also covered in detail.

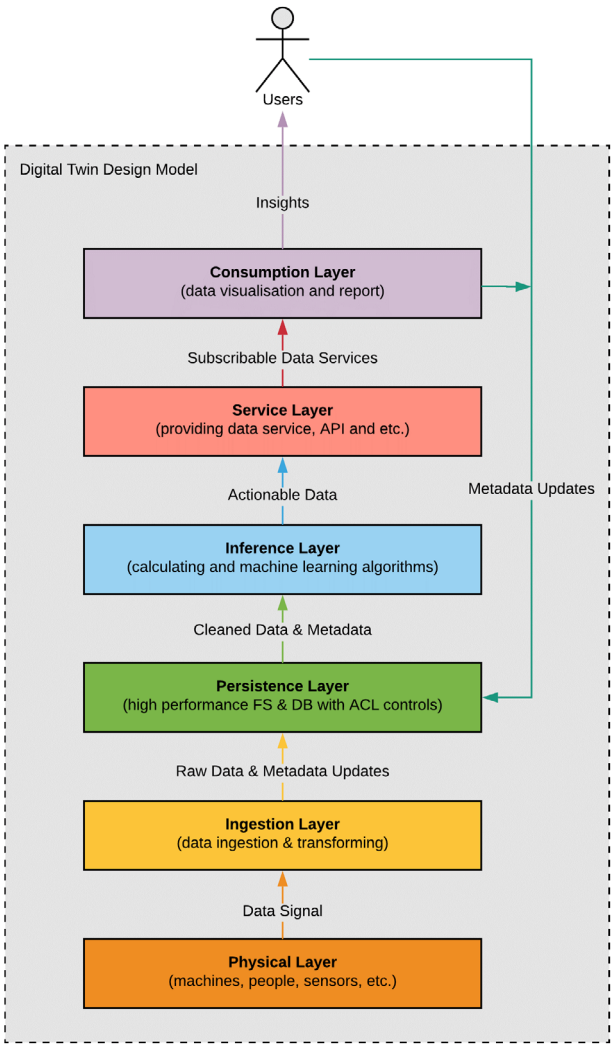


FIGURE 2 Six-layer digital twin model with upstream and feedback flows

4.1 | Physical layer

At the bottom of the six-layer digital twin design model, the physical layer is responsible in collecting all data from physical twin and its encompassed dimensions and converting it into any signal that can be recognized by computer programs, namely, the digital twin can utilize (Figure 3). It connects the physical twin and the digital twin by completely capturing all attributes and statues of the physical assets, which are to be digitally replicated. For example, a sensor of a machine that detects and sends the temperature to a data ingestion application is categorized a component of the physical layer. Thus, any entities that are generating or collecting data from the physical assets of a physical twin and its encompassed dimensions, and then sending the data to ingestion programs of the digital twin, are considered as components of physical layer.

4.2 | Ingestion layer

The ingestion layer (Figure 4) is the second layer in our model that sits immediately on top of the physical layer. The components of this layer are responsible for extracting, transforming, and transferring data or metadata receives from different sources, and loading them into the persistence layer (the next layer).

In this layer, raw data that comes from the physical layer will first be extracted from interfaces that the physical layer components provided. Afterward, data will be transformed into certain formats for storing purposes. For example, depending on different data sources, the format of the extracted data might be different such as CSV, XML, JSON, or directly from ODBC/JDBC. To load the received data with different formats, the applications in the ingestion layer must be able to decode the data and transform them into a required format that is utilized in the persistence layer.

Besides raw data, it is necessary to emphasize that the metadata of a digital twin must also be ingested in this layer. For example, in a manufacturing plant, the entity hierarchies that represents the relationships between machines, devices and functional areas are one type of metadata

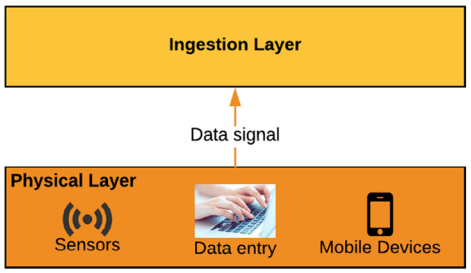


FIGURE 3 Digital twin physical layer

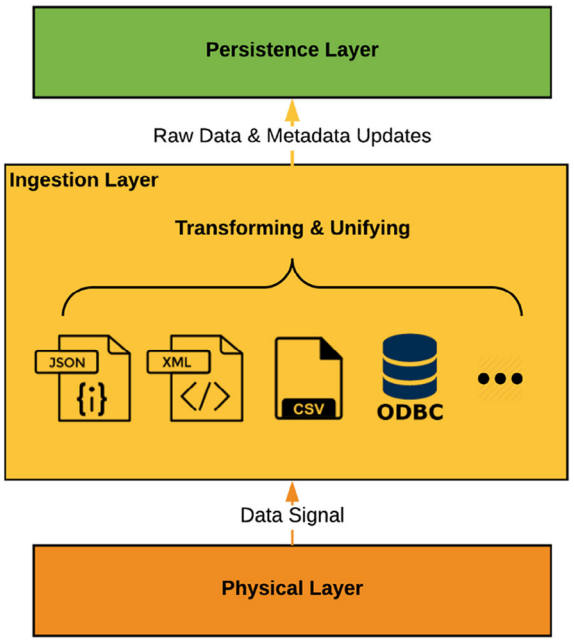
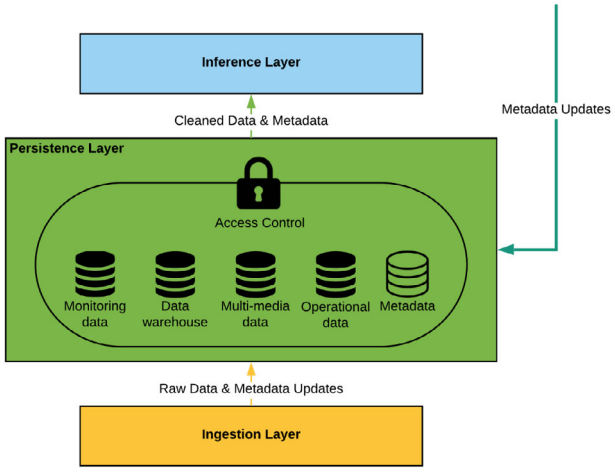


FIGURE 4 Digital Twin Ingestion Layer

FIGURE 5 Digital twin persistence layer



of their digital twin. However, they can be varied over time due to the changes such as machine model upgrade. Therefore, this metadata has to be ingested and updated frequently to ensure these changes are always correctly replicated by the digital twin, in order to appropriately guide the data association processes happened in the inference layer.

4.3 | Persistence layer

The persistence layer (Figure 5) is considered to be the storage warehouse of all data and metadata for its digital twin, but this is not its only functionality. The data persistence methods and architectures can heavily impact on the overall performance, as a result compromising user experiences and usability of the entire digital twin system. Security concerns are also raised in this layer, which should be implemented in a way that allows permission schemes to also be applicable to any digital twins at the same level in a digital twin hierarchy if exists. In addition, it is also important to involve some personal identification techniques such as the methods proposed by Ogiela and Ogiela to ensure the authorization of this layer.[29,30]

As stated in the section big data ecosystem 3.2, big data ecosystems are usually leveraged to achieve the goal of the persistence layer because of its scalable performance on large volume and high velocity data. More importantly, selecting the correct data tools based on the nature of the data stored to maximize the I/O speed is critical to the performance of the persistence layer. For example, the tools for storing time-series data should have different optimizing techniques compare with user profiles data. This is not only because of their different formats, but also accounts for the usage frequency and the size of each response per requests.

It is also important to develop an organized access control list scheme in the storage layer. This is especially important for a multitenancy digital twin system. Although a digital twin is supposed to have a "360° view" of its physical twin, the perspectives might need to be limited for certain clients. Therefore, it is required to have the ability to restrict some users' access based on the practical business requirements, in order to keep appropriate confidentiality of data.

4.4 | Inference layer

The inference layer (Figure 6) is responsible for all types of calculations in a digital twin. This includes a wide range of data computing jobs from simple formula-based calculations to machine learning algorithms. There are four important designing objectives of the inference layer, which are:

- Automation
- Metadata-driven
- Fault tolerance
- Preaggregation for high frequently demand data

Moreover, the types of jobs that the inference layer of a digital twin are classified as follows:

- Calculation jobs that generate new data from existing data based on some formula
- Data association jobs that connecting data from different data sources based on predefined rules or semantic models, namely, the metadata.

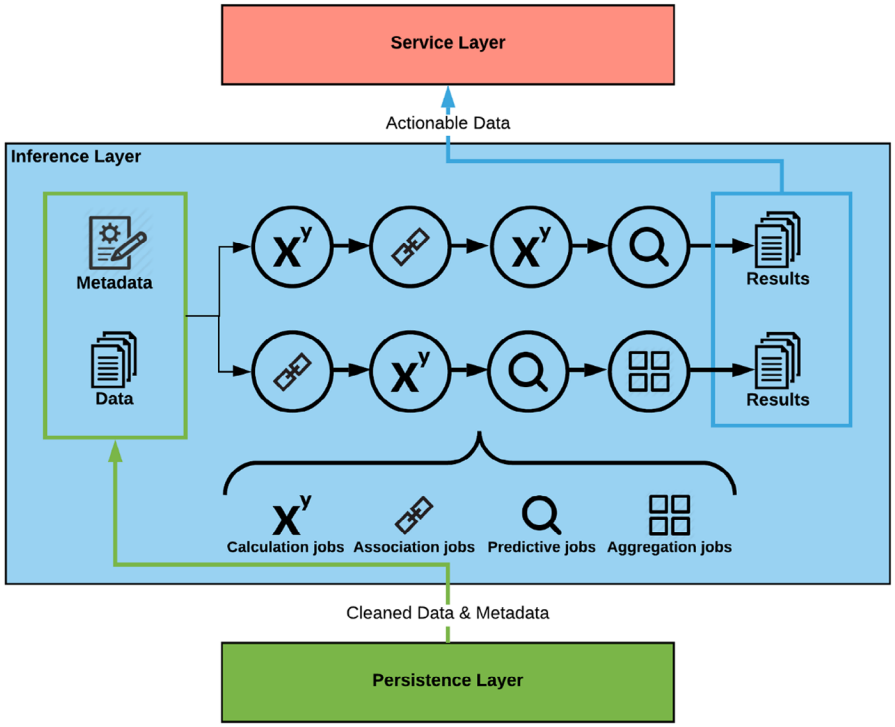


FIGURE 6 Digital twin inference layer

- Predictive jobs that utilize trained machine learning models to generate predicted results from existing data
- Aggregation jobs that aggregate any calculated/associated/predicted actionable data at a certain level to improve query performance of the digital twin for reporting purposes

(1) Automation of inference layer

To achieve the continuous intelligence of digital twin, all jobs in inference layer should be automatically triggered and executed, and in most cases in a certain order. For example, there is one inference job that is to read the raw data from the persistence layer and perform some calculations. This generates some new data, which then feeds another job that will perform machine learning-based predictions utilizing these generated data. Obviously, the predictive job is "dependent" on the first inference job. In other words, there is a strong dependency between these two jobs so that the order of executing should be guaranteed.

Scheduling jobs with certain frequencies is one solution to satisfy the dependencies. This requires a precise estimation of the duration of all jobs in a sequence. The frequency of each job is defined by its possible longest processing time. However, the "possible longest" might not be always true, for instance, there are expected down events. Another solution is to utilize data streaming in the digital twin system. This allows data to be processed step by step where each step is an application that does one type of calculation of the data. This is considered as more advanced with less latency and more reliability, because the later step will only start when it receives the data from previous one.

(2) Metadata-driven job configurations

Because jobs in the inference layer should be executed automatically, it is important to provide guidelines of the execution with configurations and parameters. This is considered a type of metadata of the digital twin. For example, there might be some parameters for machine learning algorithms and rules for the rule-based data associating programs. The metadata is supposed to be either user-defined or ingested from the ingestion layer and both are stored in the persistent layer. Moreover, this design pattern enables user to modify the metadata in order to adjust the behaviors of the algorithms in inference layer, which is the metadata updating feedback flow.

(3) Fault tolerance

To improve the robustness of the digital twin, the inference layer needs to be designed with job failure tolerance. Typically, a data store can be created in the persistence layer that only stores the most recent successful run of each of the inference jobs, which are called break points. Therefore, any jobs will be executed from the break point, and this ensured the fault tolerance. For example, if a job failed from its executing, the break point will not be updated, and then its next run will still be started using the original break point, so this is equivalent to redo the failed previous run and the current run.

(4) Preaggregation of high demand data

It is sometimes necessary to implement aggregation jobs in the inference layer. This type of jobs do not generate any new dimensions of existing data, but aggregate and generalize them into higher levels in order to improve query and retrieval performance of the digital twin. For example,

there is a predictive job that continuously produces predictive results every second, which satisfies the day-to-day monitoring purpose, such as displaying the values as a daily trend. However, if this predictive result is frequently needed in a daily, weekly and monthly summarized report, it is then necessary to implement an aggregation job that preaggregates these results to directly serve the high-level reports. Therefore, the performance of "compute once" is considered more advanced than "compute on every request."

4.5 | Service layer

The service layer consists of various types of interfaces. It also includes constraints of accessibility of the data in persistence layer to the applications in the consumption layer. After the inference layer, there will be different types of data in the persistence layer such as raw data, metadata, derived data, predictive data and aggregated data, which together describes the physical twin in a 360° view. However, it is still necessary to establish multiple analytical models which have distinct view perspectives. That are, the data services provided by the service layer APIs will have predefined perspective of the physical twin that can be subscribed by the data consumers. In addition, because each of the analytic model provides data for only one topic of the digital twin with access control, it is also considered the first security gateway of the confidential data to certain client. In other words, an API end point will verify the credentials of the data subscription before the actual request is honored. For example, a data consumer application may need both raw data and derived data, whereas a reporting tool only requires aggregated data. In this case, the trending data and the aggregated reporting data are two different analytical models with different focuses that are published by different API end points as data feeds. These feeds can also be configured with different user authorizations.

Sometimes a standalone API server is necessary in the service layer when data in the persistence layer are stored in different databases with different formats and query interfaces for storage and performance optimizing purposes. For example, time-series data and aggregated reporting data are not likely to be stored in a same database for the elaborated reasons. In this scenario, A standalone API server can implement both the interfaces of the different databases and provide the encapsulated data services with a unified formatted data so that it can be utilized by consumption layer, as shown in Figure 7.

4.6 | Consumption layer

The consumption layer directly serves the users of a digital twin which can be any self-developed and third party software as shown in Figure 8. It is common to have dashboarding requirements in a digital twin project, such as monitoring, reporting, and interactive analytics. In this layer, the data services provided by the services layer are subscribed, and then the data is provided to the data consumption software such as Qlik[31] and Tableau[32] that visualize them and provide insights to the users.

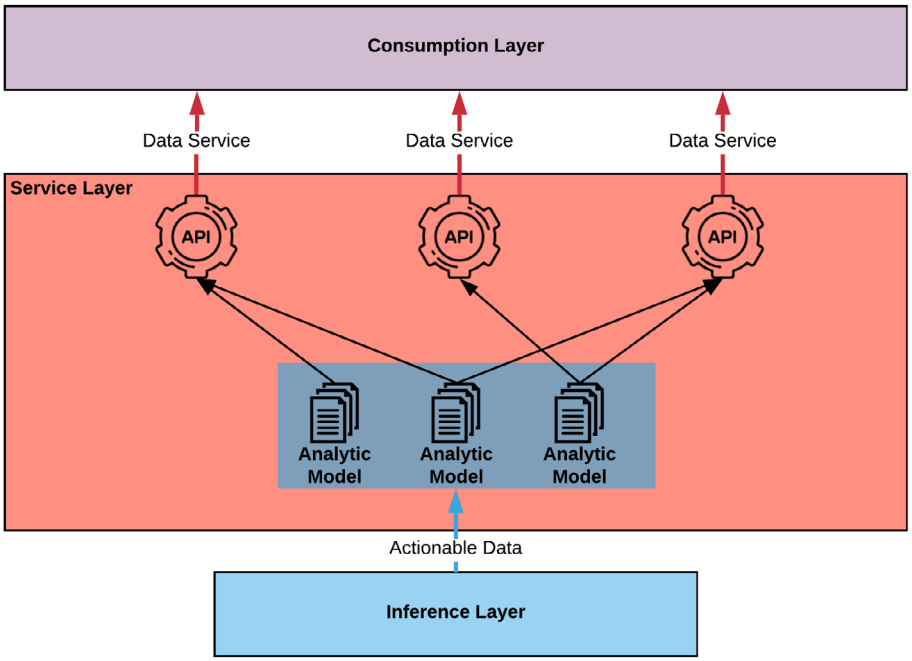


FIGURE 7 Digital twin service layer

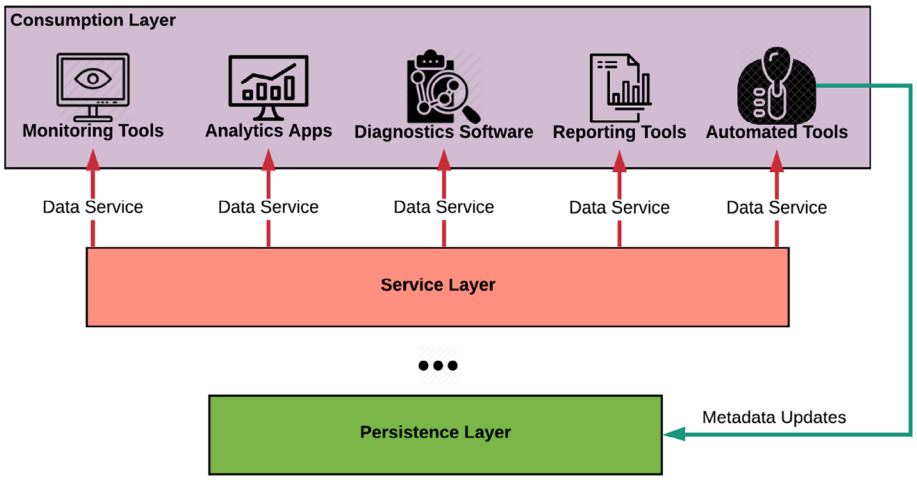


FIGURE 8 Digital twin consumption layer

It is also important to mention that when the data analytics project arrives at the fourth phase (see the section Autonomy and Scale-Expansion), the tools developed to automate the whole digital twin are also implemented in this consumption layer. The only difference between automated tools and the other data consumption tools is that the former may or may not be utilized for data visualization purposes but must have the ability to feed the updates of the metadata back to the persistence layer of the digital twin. Therefore, the full process of the six-layer digital twin model can be automated. Specifically, the data flow is started from the physical layer, and then it is reshaped following the guide of the metadata at other layers in the middle, and finally consumed in the consumption layer. The automation tools in consumption layer diagnoses potential issues from the data and feed any necessary updates of metadata back to the persistence layer, so the solution will be applied onto the physical twin to solve these problems.

4.7 | Metadata updating feedback flow

According to the six-layer digital twin model, the upstream data flows are relatively clear and straightforward. The physical layer collects data and provides the original data signal to the ingestion layer. The ingestion layer extracts, transforms and loads raw data and metadata into the persistence layer. The persistence layer maintains the data in appropriate database management systems and ensures its security. The inference layer adds values to the data utilizing various algorithms and storage it back to the persistence layer. The service layer queries the actionable data and provides the data as services to be consumed. Finally, the consumption layer consumes the data feed to present it to the end user and applications.

More importantly, an appropriate digital twin system should not only have a data upstream, it should also deliver the decisions and adjustments, either from the users or automated tools, and feed it back to the system by adding or updating metadata in persistence layer. So, the inference layer will utilize the updated metadata to guide its calculations and reasoning. For example, a user wants to increase the sensitivity of a machine learning algorithm that is predicting possible failure events of equipment. The adjustments of the algorithm parameters will be sent to the persistence layer to replace the old parameters. Then, the algorithm can start to perform predictive jobs according to the updated metadata.

The data upstream and feedback control flows are shown in Figure 2.

4.8 | Digital twin modularization

As discussed in section AI Integration, a digital twin can be modularized so that multiple digital twins can be treated as a component of a parent digital twin. This reflects the hierarchical structures of the physical twin. For example, as shown in Figure 9, a digital twin of a train will become one of the physical layer components of the digital twin of metro system that it belongs to, and the digital twin of the metro system will act as an application of the consumption layer of the digital twin of this train. More specifically, the insight data of the train's digital twin (child DT) will be ingested into the metro's digital twin (parent DT) for further analysis so that it can achieve a 360° view of the whole metro system. This includes all the trains, and is also able to utilize the data from each train to perform AI algorithms for the system. For example, the real-time operating information from the trains' digital twins will be ingested into the metro digital twin, and then these data will be utilized to reschedule, control the trains such as their real-time speeds, and generate announcement messages.

In terms of the connections between the sibling digital twins, they can exchange data from their consumption layers to assist the intelligence of themselves. In Figure 9, the digital twins of metro and streets will exchange information from each other to support passengers transferring. The

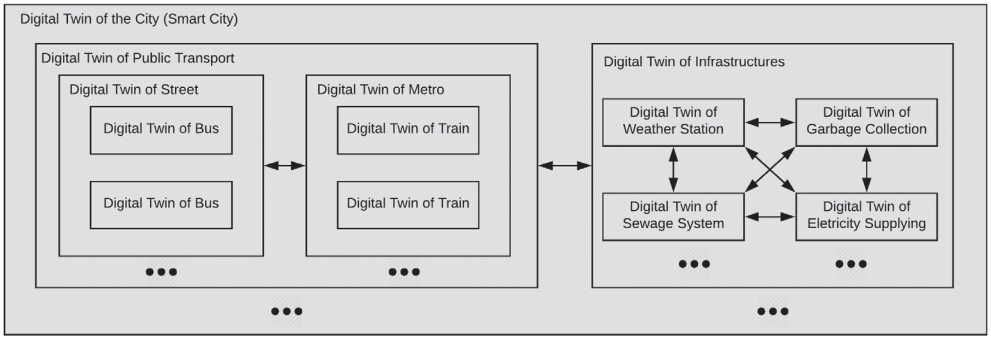


FIGURE 9 Digital twin of a city (Smart City)

digital twin of sewage system and weather station will collaborate to prevent flooding in the city. The digital twin of the city infrastructure will pass on information from the digital twin of metro to the digital twin of electricity supply to trigger any emergency response when there is unexpected spike in public transportation traffic.

Thus, the expandable feature of digital twins can be generalized and applied to any physical objects. For example, a digital twin of a car is driving on a street with digital replication to manage the traffic. There are also multiple digital twins not only for the streets, but also other infrastructures that are relevant in this context. All of them can be digitalized and finally formed into a context-aware digital twin of a city, which is proposed recently as "Smart City".[16]

5 | INDUSTRIAL CASE STUDY

A manufacturing digital twin system with predictive maintenance[33,34] has been designed and implemented in a manufacturing company with multiple plants. In this section, we would like to briefly introduce this use case as a case study to demonstrate the validity of the proposed six-layer digital twin model. The system aims to predict and prevent equipment breakdowns of a complex manufacturing plant in both big data and real-time environment. With the advances of modern sensors in terms of sensing capabilities, high-density sensor installation is implemented to provide the seamless monitoring of various equipment. Therefore, equipment sensor data are generated by the physical layer. The data workflow of our system includes: ingest sensor data in the ingestion layer, store the preprocessed data in the persistence layer with a suitable format, design, and implement predictive model in the inference layer, and analyse and deliver the predicted results to the engineer through dashboard in the service layer and consumption layer. When we conducted a deeper study, more issues regarding practical applications are discovered and revealed. These findings triggered a new data quality issue involves noise cleansing and missing value imputation before storing data in a central cloud cluster, and sensor selection before implementing predictive model in our inference layer. A high-level architecture diagram of the entire digital twin system has been shown in Figure 10.

In this digital twin solution, we have completely utilized open-source components which will relatively easier to be replicated without involving certain vendors. However, it is also can be seen that the components can be potentially replaced by proprietary cloud solutions or platform such as Microsoft Azure. Moreover, the figure below only showed the critical components that play important roles in the six-layer digital twin system. In other words, the entire system need a large number of other services to be functional, such as Apache Oozie to scheduling and orchestrating the workflows and Apache Zookeeper that coordinates all these components working together.

Broadly, the intelligent features composing our digital twin system are summarized as follows.

5.1 | Remove noisy sensors

This step is to preliminarily weed out the sensors which do not provide accurate information and would degrade the model performance. The sensors whose variables have high percentage of noisy points or missing values are all defined as noisy ones and need to be removed, since the information from these variables is no longer reliable. Further repairments of these sensors based on unreliable information are still unreliable.

5.2 | Noise detection

Unlike the public dataset or the data collected from laboratory simulation, noise is ubiquitous in the real industrial IoT sensor data. Removing or repairing the noise enables us to develop more accurate models. Noise and anomalies both belong to the outliers, referring to the points which

FIGURE 10 A sample architecture of six-layer digital twin in our production environment

deviates from the normal points. They are related but distinct in concept and have totally different meanings during manufacturing process in the context of industrial IoT. In manufacturing monitoring data, an outlier may have two interpretations, and it can be either noise or significant anomaly. Noise may appear as the inaccurate data resulted from sensor errors, while anomalies can be items of interest caused by abnormal equipment conditions. Therefore, new challenges are introduced to noise detection for IIoT-based intelligent monitoring systems, which is to precisely extract noise in the presence of significant anomalies, and preserve the anomalies at the same time. To address this issue, our project provides one possible solution based on sliding window contrast cue.[35] Extensive experiments are conducted to prove that IoT data after noise cleaning can detect the faults earlier than the raw data, which is extremely important for smart decision making for remedial maintenance.

5.3 | Missing value imputation

This component is responsible for imputing the missing points with the most appropriate values and providing a complete dataset to be analyzed in the integrity-required applications or stored in the data warehouse. Common mode failures occur when a single event leads to the loss of data from multiple sensors. This is a commonly occurring scenario in the IoT-enabled smart manufacturing monitoring systems. Our project designs an iterative imputation framework using multiple segmented gap iteration to provide more precise results for IIoT sensor data with large gaps.[36]

After the sensor data cleansing, the raw data has been transformed into an analysis-ready phase. To detect different types of faults, separate models build upon different sensor subsets. The optimal number of sensors need to be determined to make sure all the valuable information is collected in an appropriate way. If only few sensors are installed to monitor the machine, the prediction results will be inaccurate due to the limited data. In contrary, the large number of sensor inputs will distort and obscure the outputs. Regarding to one certain fault, only the responsible sensors are identified and selected as the inputs to build a specialized model. In this way, all the sensors will be divided into several blocks to monitoring different parts of our system independently.

5.4 | Fault detection

Based on one certain sensor subset as data input, the corresponding type of fault can be detected effectively. A MapReduce-based distributed PCA model is utilized in our system for fault detection and diagnosis. In a large-scale manufacturing system, not all kinds of failure data are

accessible, and the absence of labels precludes all the supervised algorithms in the predictive phase. In addition, our system takes advantage of some of the characteristics of PCA such as its ease of implementation on Spark, its simple algorithmic structure, and its real-time processing ability.

5.5 | Contribution analysis for root cause detection

When the fault is detected by the monitoring model, contribution analysis of all the incorporated variables is required to track the root sensors that cause the issue. Thus, engineers can easily find the machines in troubles and take actions timely to prevent further damage to the whole production system.

5.6 | Sensor fault or machine fault

Further analysis on the root sensors can be conducted to distinguish the true machine faults from the possible sensor faults. Sometimes the malfunctions happen in sensors instead of the monitored machine, in which case something wrong going with the sensor itself will be mistaken for the machine faults. Our system uses contribution analysis results to classify whether a fault belongs to machine or sensor. When a machine-related fault occurs, multiple sensors should be affected, which will be reflected in their measurement readings. Accordingly, all these multiple responsible sensors should be identified as ones which contribute to the fault. Otherwise, if the detected fault is only contributed by one sensor, and the rest of the responsible sensors behave normal, this means the problem happens in this sensor itself. According to the analysis results, engineers can arrange the most effective measures to reduce the human and economic cost.

5.7 | Outcomes of the digital twin system

All the intelligent features above-mentioned were implemented and endorsed by a global chemistry manufacturing company, and these features are achieved based on the six-layer digital twin architecture which we proposed in this article. The benefits that we obtained from having such a system includes but not limited as follows:

- Minimized the manual works because all the procedures such as data ETL and machine learning predicting are automated.
- Moved data analytics to the next level by real-time streaming techniques.
- Integrated the power of machine learning in the system to quickly detect and resolve complicated issues without human involved.
- Be able to predict potential problems and either solve it automatically (by metadata updating feedback flow) or notice people in charge (physical issues such as equipment ageing) before it happened.

By knowing issues before it will cause trips (4–12 days in advance), the system helped the business to avoid huge potential loss.

6 | CONCLUSION

In this article, we first propose a data analytic maturity model, which consists of four phases with ordered activities. It shows that any data analytic projects need to be gradually developed from foundations to powerful AI algorithms. The efforts and time spent on routine will create exponentially increase in business values. The digital twin starts in phase two which immediately follows the event that the big data infrastructure is established. It is started by shallowly replicating the characteristics, features and statues of its physical twin, and then dive deeper to copy its behaviors, which is achieved by AI technologies, typically machine learning models.

In terms of the digital twin architecture, we proposed the six-layer digital twin model to classify standardize the components of a digital twin. That is, the data signal is generated by the physical layer, ingested by the ingestion layer, stored in the persistence layer, calculated in the inference layer, provided by the service layer as data services and consumed in the consumption layer where finally provides insights to users. We also found that the metadata updating feedback flow is a critical required feature of an autonomous digital twin, which enables the ability to react to any changes such as machines replacement and business rules. To validate our digital twin architecture, a reference big data ecosystem is provided, which integrates all the layer as a practice solution.

The specific standards and details of a hierarchical digital twin with modularization is one of the future works. Further research is required to formulate the process of constructing multiple digital twins into a nested one, such as the protocol that allows digital twins talk to its parents, offspring and siblings. This will bring the concept of "Smart City" to reality, as well as any other context-aware autonomous super digital twins.

ORCID

Longquan Tao https://orcid.org/0000-0002-5398-2429

REFERENCES

1. Boschert S, Rosen R. Digital twin—the simulation aspect. *Mechatronic Futures*. New York, NY: Springer; 2016:59-74.
2. Främling K, Holmström J, Ala-Risku T, Kärkkäinen M. Product agents for handling information about physical objects. Report of laboratory of information processing science series B, TKO-B; 03, 2003:153.
3. Knapp G, Mukherjee T, Zuback J, et al. Building blocks for a digital twin of additive manufacturing. *Acta Materialia*. 2017;135:390-399.
4. Qi Q, Tao F, Zuo Y, Zhao D. Digital twin service towards smart manufacturing. *Proc Cirp*. 2018;72:237-242.
5. Rosen R, Von Wichert G, Lo G, Bettenhausen KD. About the importance of autonomy and digital twins for the future of manufacturing. *IFAC-PapersOnLine*. 2015;48(3):567-572.
6. Tao F, Cheng J, Qi Q, Zhang M, Zhang H, Sui F. Digital twin-driven product design, manufacturing and service with big data. *Int J Adv Manufactur Technol*. 2018;94(9-12):3563-3576.
7. de Moura RL, Ceotto LD, Gonzalez A. Industrial IoT and advanced analytics framework: an approach for the mining industry. Paper presented at: Proceedings of the 2017 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV; 2017: 1308-1314.
8. Madigan N. Profile: people first: mining across the globe. *AusIMM Bull*. 2018;2018:14.
9. Meyer N, Horvat D, Hitzler M, Doll C. *Business Models for Freight and Logistics Services. Technical Report, Working Paper Sustainability and Innovation*. Karlsruhe, Germany: Fraunhofer Institute for Systems and Innovation Research ISI; 2018.
10. Zhou C, Luo H, Fang W, Wei R, Ding L. Cyber-physical-system-based safety monitoring for blind hoisting with the internet of things: a case study. *Automat Construct*. 2019;97:138-150.
11. Verner I, Cuperman D, Fang A, Reitman M, Romm T, Balikin G. Robot online learning through digital twin experiments: a weightlifting project. *Online Engineering & Internet of Things*. New York, NY: Springer; 2018:307-314.
12. Botkina D, Hedlind M, Olsson B, Henser J, Lundholm T. Digital twin of a cutting tool. *Procedia Cirp*. 2018;72:215-218.
13. Tao F, Zhang M, Liu Y, Nee A. Digital twin driven prognostics and health management for complex equipment. *Cirp Annals*. 2018;67(1):169-172.
14. Redelinghuys A, Basson A, Kruger K. A six-layer digital twin architecture for a manufacturing cell. Paper presented at: Proceedings of the International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing; 2018:412-423; Springer, New York, NY.
15. Lasi H, Fettke P, Kemper HG, Feld T, Hoffmann M. Industry 4.0. *Bus Inf Syst Eng*. 2014;6(4):239-242.
16. Alam KM, El Saddik A. C2PS: a digital twin architecture reference model for the cloud-based cyber-physical systems. *IEEE Access*. 2017;5: 2050-2062.
17. Mirian A, Ma Z, Adrian D, et al. An internet-wide view of ICS devices. Paper presented at: Proceedings of the 2016 14th Annual Conference on Privacy, Security and Trust (PST), Auckland, New Zealand; 2016:96-103; IEEE.
18. Sagiroglu S, Sinanc D. Big data: a review. Paper presented at: Proceedings of the 2013 International Conference on Collaboration Technologies and Systems (CTS), San Diego, CA; 2013:42-47; IEEE.
19. Miloslavskaya N, Tolstoy A. Big data, fast data and data lake concepts. *Proc Comput Sci*. 2016;88(300-305):63.
20. Nassiri P, Monazzam MR, Golbabaei F, et al. Application of universal thermal climate index (UTCI) for assessment of occupational heat stress in open-pit mines. *Industrial Health*. 2017;55(5):437–443.
21. Chen M, Mao S, Liu Y. Big data: a survey. *Mob Netw Appl*. 2014;19(2):171-209.
22. White T. *Hadoop: The Definitive Guide*. Newton, MA: O'Reilly Media, Inc.; 2012.
23. Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters. *Commun ACM*. 2008;51(1):107-113.
24. George L. *HBase: The Definitive Guide: Random Access to Your Planet-Size Data*. Newton, MA: O'Reilly Media, Inc; 2011.
25. Zaharia M, Xin RS, Wendell P, et al. Apache spark: a unified engine for big data processing. *Commun ACM*. 2016;59(11):56-65.
26. Gao L, Zhang C, Sun L. RESTful Web of Things API in sharing sensor data. Paper presented at: Proceedings of the 2011 International Conference on Internet Technology and Applications, Hubei, China; 2011:1-4; IEEE.
27. Wang L, Orban P, Cunningham A, Lang S. Remote real-time CNC machining for web-based manufacturing. *Robot Comput Integrat Manuf*. 2004;20(6):563-571.
28. Nguyen KA, Do P, Grall A. Multi-level predictive maintenance for multi-component systems. *Reliab Eng Syst Saf*. 2015;144:83-94.
29. Ogiela L, Ogiela MR. Bio-inspired cryptographic techniques in information management applications. Paper presented at: Proceedings of the 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), Crans-Montana, Switzerland; 2016: 1059-1063.
30. Ogiela MR, Ogiela L. Cognitive keys in personalized cryptography. Paper presented at: Proceedings of the 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA), Taipei, Taiwan; 2017:1050-1054.
31. Troyansky O, Gibson T, Leichtweis C. *QlikView Your Business: An Expert Guide to Business Discovery with QlikView and Qlik Sense*. Hoboken, NJ: John Wiley & Sons; 2015.
32. Murray DG. *Tableau Your Data!: Fast and Easy Visual Analysis with Tableau Software*. Hoboken, NJ: John Wiley & Sons; 2013.
33. Yu W, Dillon T, Mostafa F, Rahayu W, Liu Y. A global manufacturing big data ecosystem for fault detection in predictive maintenance. *IEEE Trans Ind Inform*. 2019;16(1):183-192.

34. Yu W, Dillon T, Mostafa F, Rahayu W, Liu Y. Implementation of industrial cyber physical system: challenges and solutions. Paper presented at: Proceedings of the 2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS), Taipei, Taiwan; 2019:173-178.
35. Liu Y, Dillon T, Yu W, Rahayu W, Mostafa F. Noise removal in the presence of significant anomalies for industrial IoT sensor data in manufacturing. *IEEE IoT J*. 2020;7(8):7084-7096.
36. Liu Y, Dillon T, Yu W, Rahayu W, Mostafa F. Missing value imputation for industrial IoT sensor data with large gaps. *IEEE IoT J*. 2020;7(8):6855-6867.

How to cite this article: Mostafa F, Tao L, Yu W. An effective architecture of digital twin system to support human decision making and AI-driven autonomy. *Concurrency Computat Pract Exper*. 2020;e6111. https://doi.org/10.1002/cpe.6111