https://doi.org/10.1016/j.rcim.2020.102042

Development of an edge computing-based cyber-physical machine tool

Jian Zhang[1, 2], Changyi Deng[2, *], Pai Zheng[4], Xun Xu[3], Zhentao Ma[2]

1. School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, PR China

2. Institute of Systems Research, China Industrial Control Systems Cyber Emergency Response Team, Beijing 100040, PR China;

3. Department of Mechanical Engineering, The University of Auckland, Auckland, New Zealand

4. Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong, PR China

Abstract:

Digital twin is a virtual model that represents physical entities in a digital manner. By leveraging means of data to simulate the behavior of physical entities in the real environment, the functions of physical entities can be optimized and expanded, through virtual and real interaction feedback, data fusion, decision making, and optimization. Despite numerous researches on digital twin concept and its applications, scarcely any discusses about the computation efficiency of the twin established. In order to shorten the latency of mapping and reduce the high computation workload in the cloud, this paper develops a cyber-physical machine tool based on edge computing techniques, to realize remote sensing, real-time monitoring and scalable high-performance digital twin application. Furthermore, a novel edge computing algorithm is proposed to detect the abnormality of the edge data from two aspects: the unary outliers of the edge data itself and the multivariate parameter correlation among edge devices. The effectiveness of the application platform of the cyber-physical machine tool developed is verified by the prototype system and edge algorithm experiment.

Key words: Digital Twin; Mapping; Edge Computing; Machine Tool

1.Introduction

To date, IoT, big data, and artificial intelligence play a significant role in Cyber-Physical Systems (CPS) [1-2]. One of their purposes is to bring the next generation of information and communication technology to achieve mapping between physical space and virtual space [3-4]. Cyber-Physical Production Systems (CPPS) are representation of CPS in a production environment. As a key component of CPPS, the next generation machine tool has been proposed named Cyber-Physical Machine Tools (CPMT) with Industry 4.0 which in-depth integration of machine tool, edge computing, networking and digital twin (DT) [1]. A CPMT will have its digital space of machine tool, as the digital twin with networking capabilities and computational, it allows the establishment of a real-time feedback loop, which can affect the calculation process and vice versa [5]. CPMT will have a machine tool digital space as a digital twin with computing and networking capabilities, allowing real-time feedback loops to be established where machining process and calculation results can feedback control via digital twin.

DT is a core technique for CPS with various definitions [6-12], and the most widely accepted one was originally given by Glaessegen and Stargel in 2012: "*digital twin means an integrated multiphysics, multiscale, probabilistic simulation of a complex product, which functions to mirror the life of its corresponding twin*" [13]. CPS emphasizes the real-time, dynamic information feedback and circulation process between the physical world and the digital world. Digital twin as a key technology different from CAD and IoT, the former focuses on the interaction of digital world, and the latter focuses on the perception of physical world. However, DT acts on the entire life cycle of manufacturing with the two-way interaction between physical world and digital world, which establishes virtual models for physical entities, and simulates the behavior of physical entities in the real environment [14]. Physical entities can be more "intelligent" to optimize their real-time behavior through virtual models. Meanwhile, the characteristics of the virtual model can be more "real" to show physical entity.

Digital twin and edge computing are key technologies to build a CPMT. Existing studies scarcely discuss the computation efficiency of the twin established for the machine tool. In addition, how to short the latency of mapping and reduce the high computation workload in the cloud, has not yet been analyzed. Therefore, in this paper, a cyber-physical machine tool (CPMT) based on edge computing and digital twin techniques has been developed, to realize remote sensing, real-time monitoring and scalable high-performance digital twin application, and the importance of this gap also has been analyzed and evaluated. However, the data generated by edge devices is affected by network bandwidth and cloud. Edge computing is used to transform the edge data into a mirror of the virtual machine tool with MTConnect and reduce cloud modeling pressure. Furthermore, a novel edge computing algorithm is proposed to detect the abnormality of the edge data from two aspects: the unary outliers of the edge data itself and the multivariate parameter correlation among edge devices.

According to the mentioned issues above, the main contributions of this paper are as follows:

- Proposed a three-tier architecture of Cyber-Physical Machine Tool based on edge computing.
- Explored edge computing technique to improve the accuracy and capabilities of virtual machine tools.
- A novel edge computing algorithm is proposed to shorten the latency of mapping and reduce the high computation workload in the cloud.
- A prototype system is developed to realize virtual processing and simulation, and the effectiveness of the application platform of CPMT developed is verified by the prototype system and edge algorithm experiment.

The next section in this paper presents related works. Section 2 discusses the challenges DT. In addition, Concept of CPMT is introduced in Section 3. Section 4 demonstrates the edge computing detection algorithm for CPMT. In Section 5, a prototype development of CPMT is designed, Physical space and cyber space are presented by a machine tool, MTConnect agent and adapter. In Section 6, a case study of CPMT is presented and its efficiency is evaluated by multiple application scenarios. Section 7 concludes the research and future works.

2. Related work

To give an outline of the background and further discover the research gaps, this section introduces all-around review of the related researches on challenges of DT including DT in shop floor and edge computing detection algorithm for CPMT.

2.1 DT in shop floor

NASA has applied DT to the health maintenance and support of aerospace aircraft which achieved good results [15-16]. Schroeder explores automation techniques with DT and proves to be very useful in data exchange between different systems [17]. Tao Fei proposed the concept of DT shop-floor (DTS) [18], the DTS architecture, system composition, operation mechanism and key technologies were given in detail [19]. Aitor Moreno proposed a method to build a DT for punching machine which was used to develop an interactive programming application of CNC machining [20]. In order to make a typical manufacturing equipment more intelligent, a study about DT is conducted for CNC machine tool and a mapping method between machine tool and digital space was proposed [12]. Deng et al. designed a health monitoring system for a CPMT and two methods proposed were used to data cleansing and energy-saving [5]. Many leading Industrial Enterprises such as PTC, Siemens, GE, and ANSYS have also developed various applications with the guidance of DT concept [21].

In a word, DT is mostly used for fault diagnosis, predictive maintenance and performance analysis. Existing research of DT is always used for the design, operation and maintenance of complex systems (e.g. Aero-engine maintenance, Automotive production, Wind turbines, etc.) and is rarely devoted to exploring the application of DT in unit-level equipment, such as CNC machines.

2.2 Edge computing

Edge computing techniques aim to increase the computation efficiency of the twin established and reduce the pressure on cloud. The edge devices collect data with a certain frequency and sends the data to the corresponding data receiving end [22]. The data receiving end will receive one or more sets of observation sequences that have a strict sequence in time. These time series data accurately record the real-time changes of a specific parameter and reflect the trend and law of the parameter within a certain time range. However, in the actual data collection scenario, the edge devices always have some abnormalities in the process of data acquisition and transmission. The literature [23] conducted a related research on data anomaly detection for the actual edge data set, which is very difficult to get high quality data through edge devices.

Nowadays, edge computing detection algorithm mainly includes 6 types: statistical-based detection method [24], distance-based detection method [25], density-based detection method

[26], neural network-based method [27], support vector machine-based method [28] and cluster analysis method [29]. Time series data has some special properties, anomaly detection algorithm need to consider its characteristics. Most of the methods are based on pattern recognition and clustering for anomaly detection in the field of time series data [30]. Vlachos et al. [31] proposed a nonparametric method for accurate periodic detection and introduced a new algorithm of periodic distance for time series. Cattivelli et al. [32] proposed a distributed detection algorithm to detect a known deterministic signal under Gaussian noise based on diffusion strategies. However, in many applications of interest, the measurements taken by spatially distributed nodes are statistically dependent. Other researchers have considered dependent observations using Gaussian Markov Random Fields [33], [34] to design a Neyman-Pearson detector in a centralized scenario. However, the design of distributed detection algorithms with dependent measurements in a decentralized scenario deserves more investigation. Fujimaki et al. [35] proposed a novel anomaly detection system which mainly used the correlation vector regression and data auto-regression for anomaly detection. Cai et al. [36] proposed a new time series data anomaly detection algorithm by constructing distributed recursive computing strategy and k-nearest neighbor fast selection strategy.

However, those methods mainly aim at detecting abnormality monitoring work for a single edge source data. In the IoT, there are often numerous “known” correlations between different edge source data which may reveal a certain rule of data trend, and can help us effectively identify the corresponding data anomalies, to improve the accuracy of the twin established.

To address the above issues, physical machining process of machine tool needs to map virtual space in real time with DT. In order to improve accuracy of mapping, a three-tier architecture of an edge computing-based CPMT is proposed in section 3.

3. Cloud-edge computing-based CPMT

3.1 Overall architecture of CPMT

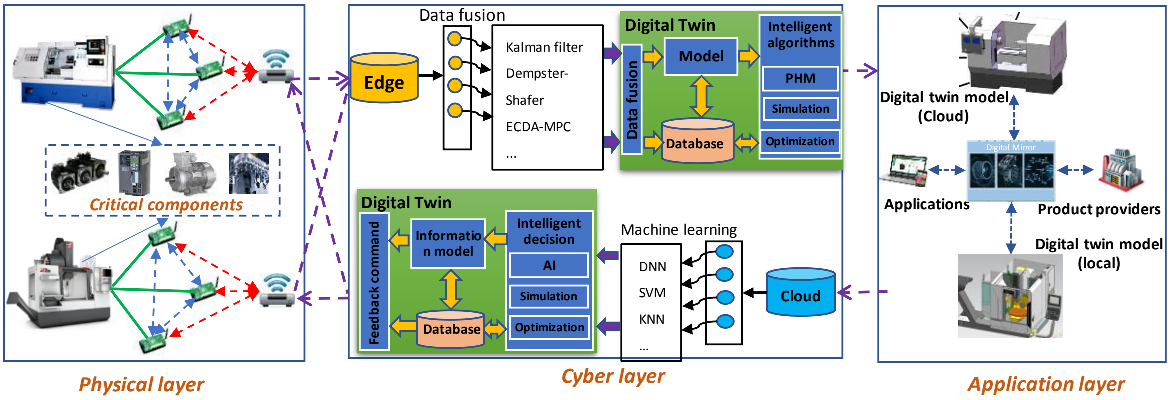

Figure 1. Architecture of an edge computing-based CPMT.

In our previous research, CPMT has been studied in vertically and horizontally integrated machine tools [1], standardized information modeling with OPC UA and MTConnect [37], and augmented reality (AR) for improving the efficiency during the machining process [38]. However, the role of edge computing has not discussed in detail. This paper focused on the efficiency of digital twin modeling based on the preliminary work and complemented the lack of cloud-edge computing. Digital twin and edge computing are the key methodology to build a CPMT, which is a digital model in the digital world that reflects the authenticity of physical prototypes. The performance of the equipment is tested and evaluated prior to physical prototype manufacturing through comprehensive simulation of multiple fields and performance attenuation simulation of the equipment. Improving its design flaws can shorten its design improvement cycle. CPMT is integration of machine tool, CNC controller and hydraulic multi-domain system, which can map the entire life cycle of machine tool. This provides powerful analytical decision support for design simulation and predictive maintenance of machine tool. The main goal of the high-performance DT application for CPMT is to develop a DT application with MTConnect standard and edge computing, which is used to monitor the processing and operation status of CNC and visualize its flow data, and edge computing is used to transform the edge data into a mirror of the virtual machine tool to reduce cloud modeling pressure. Architecture of an edge computing-based CPMT is shown in Figure 1, which includes three layers:

- *Physical layer*: Hardware includes CNC machine tools, sensors, computer and a microcontroller board. The agent interprets the data from the sensor and formats the data into an XML format that conforms to the MTConnect standard. After the edge devices deployed on various manufacturing units, it can be to build an intelligent machine tool that can sense and adapt to different production tasks. For example, when a processing task is scheduled to a machine tool, which can select suitable cutting tools to work out a production schedule based on the required quality standards and specifications.

- *Cyber layer*: A mirror of machine tool is built in this layer. Edge computing detection algorithms can work to improve the computation efficiency of the twin established. After the complex data model is calculated in the cloud, it is feedback to the physical layer through DT layer to achieve closed loop. DT layer also includes MTConnect data model [39], STEP data model [40] and XML data model [41]. With multi-source data gathered, data fusion is used to pick up feature information to watch out the status of a single machine tool. The information model is responsible for handling various heterogeneous data into a standardized format. In turn, the formatted scheme can be further used for advanced decision by Deep Neural Networks (DNN), Support Vector Machine (SVM) and k-nearest neighbors (KNN).
- *Application layer*: The MTConnect client can be used to check for typical faults and friction of the machine, which takes data from the agent and uses the acquired twin data to draw graphics in real time, reflecting the machine's processing and state. The application is developed as an end-user interface for real-time visualization of various collected data and processing status from the manufacturing frontier (e.g., the shop floor).

3.2 Modeling of CPMT

In the manufacturing world, CNC machine tool as a typical manufacturing equipment, can be considered as one of shop floor production cells, of which milling and grinding operations are performed by embedding highly integrated fieldbuses, PLCs, sensors and actuators [42]. Most components currently provide some types of monitoring or diagnostic information interface, but different vendors use different communication interfaces and protocols to make it difficult to collect and unify information. The real-time requirements of industrial communication systems may also limit the amount of data transmission. A versatile and flexible transport format is required to tailor the different types of diagnostic data. As a new communication standard, MTConnect can eliminate the data format of multiple obstacles and realize the compatibility and transmission of product data of multiple manufacturers [42]. As a middleware standard with the ability to pass data, it can not only seamlessly connect with existing operating standards (rather than replace them), but also convert from existing data transfer formats to XML-based data formats.

For smart devices, the collaboration between multiple devices (M2M) and the collaboration between the business management system and the production line (B2M) and the data between the business units (B2B) require model coordination. In this paper, the spindle of machine tool as a model instance, and the data format is modelled by MTconnect, which implements uniform definitions such as names, units, attributes, and scenarios. Figure 2 shows the basic structure of the MTConnect information model using the machine spindle motor. CNC machine tools are described as a device D1, including spindle motor, shaft and controller components, among which the spindle motor is monitored, mainly including position, speed, temperature, acceleration and other information. Each category has its corresponding data items, including timestamps, serial numbers, monitored values etc.

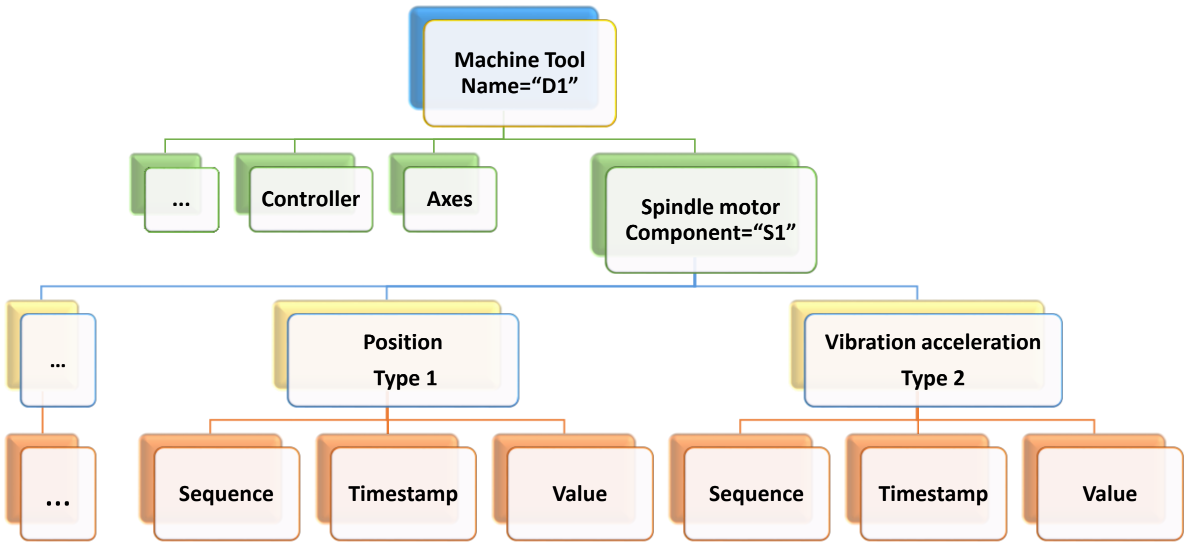

Figure 2. DT model ontology expression based on MTConnect standard.

3.3 Operation mechanism of CPMT

Altintas et al. [43] proposed virtual process systems for part processing operations, explaining various scenarios of virtual machining, including cutting force, torque, drive, stability and vibration. Yao et al. [44] proposed a loosely coupled architecture machine tool control system operating in network environment named INC. Zheng et al. [45] proposed a novel IT-driven product development model, i.e. SCOAP, which meets the adaptable design principles for product expandability and lifecycle requirement. Nevertheless, the current problem is that there still lacks a system to achieve integration of different functions.

To address it, Figure 3 reveals the operating mechanism of CPMT in the Cyber-Physical Production Systems and its role in the system. CPMT updates its own status in the cloud, then feedback to machine tool manufacturers, which improve machine design and provide machining solutions/services, to provide access for third-party value-added services. Advanced human-machine interactions for intuitive and comprehensive understanding of machine tool status and machining processes. CPMT can proactive maintenance and prevent unexpected breakdown (autonomously send alerts/emails to maintenance team when potential failures detected). Feedback to production managers including periodically send statistical production reports, Enterprise Resource Planning (ERP), and Overall Equipment Effectiveness (OEE). It also feedback machining performance to product designers and process planners to optimize design and machining process.

The virtual manufacturing business begins with the receipt of a personalized custom order by MES. MES receives the customized order from the customized system (e-commerce) to schedule the production and releases the daily production plan to CPMT. CPMT can autonomous emergency stop to prevent damage, and support process optimization: in-process adjustment of machining parameters (e.g. federates, spindle speed) through proprietary APIs.

If CPMT has an effective historical manufacturing strategy for personalized products, it directly forwards the daily production plan to the manufacturing execution system for actual production; if CPMT does not store personalization, the effective historical manufacturing strategy of the customized product generates the simulation analysis operation, the simulation analysis model are sent to the simulation analysis system for production simulation analysis, and the production simulation analysis system performs the production process according to the simulation analysis model and the data of the customized product. The results of the simulation analysis are feedback to CPMT, which receives the simulation analysis results feedback by the simulation analysis system, and the simulation analysis data that needs to be timely feedback to the user is packaged into the result of the virtual manufacturing and feedback to ERP and MES, which will be used for actual manufacturing. The execution strategy and production plan are issued to the manufacturing execution system for actual product manufacturing.

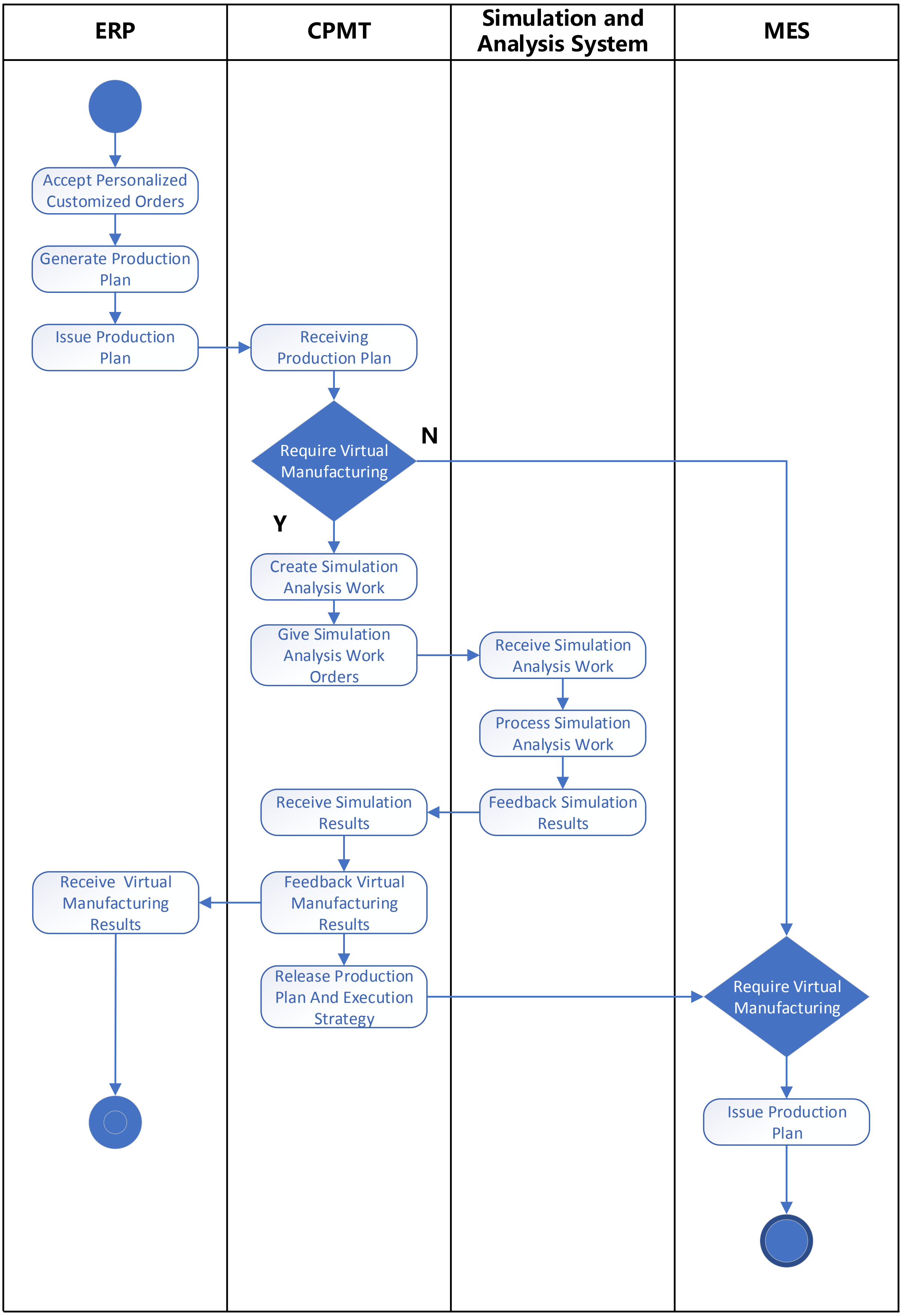

Figure 3. Operation mechanism of CPMT in CPPS.

4. Edge computing detection algorithm for CPMT

CPMT uses the mapping model to collect related data and analyses the protocol. The data acquired by various data acquisition devices is directly transmitted to the cloud computing center for data storage, and the powerful cloud computing center is used to complete the corresponding abnormality detection and data cleaning work. This approach is also known as: a centralized big data processing model based on cloud computing. As the amount of data on edge devices increases, the data generated by edge devices is affected by network bandwidth and cloud. Therefore, the existing centralized big data processing model needs to be adjusted accordingly, and part of the computing tasks of the cloud computing model are migrated to the edge device, which reduces the computing load of the data center while slowing down the network bandwidth pressure.

Edge computing detection algorithm for CPMT (ECDA_CPMT) proposed in this paper will detect the abnormality of the edge data from two aspects: the unary outliers of the edge data itself and the multivariate parameter correlation among the edge devices. Then the data fusion processing is performed on the two different detection results to complete the final multi-source edge data anomaly detection, as shown in Figure 4.

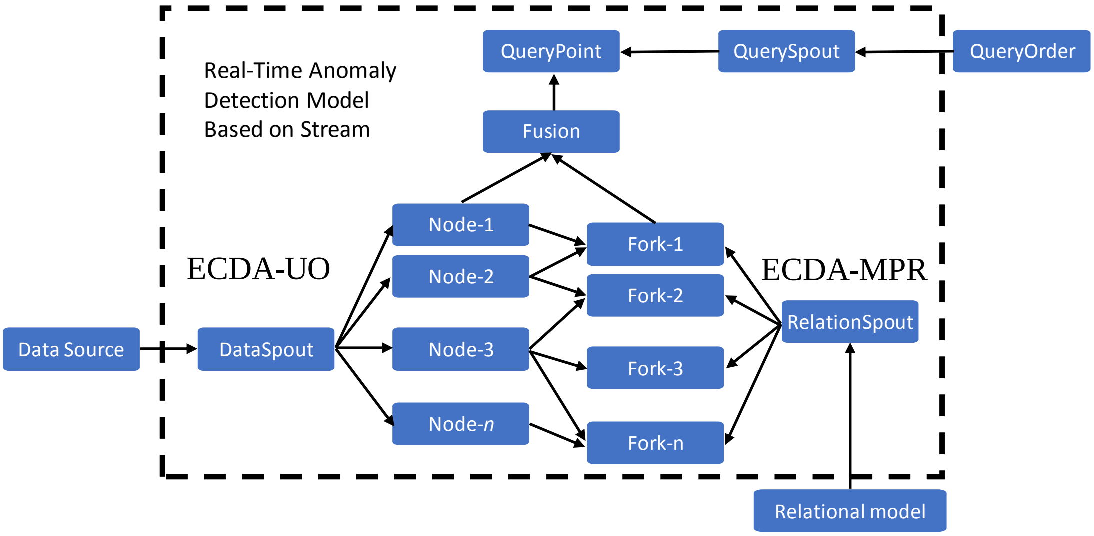

Figure 4. Edge computing detection algorithm for CPMT.

DataSpout receives the collected data and sends it to each node to check the timing continuity of the single edge data. If the data received with Data Spout is large and a wide variety of the parameter type, the data can be divided by the data partitioning module, and the divided data is transmitted to the fork of edge computing detection algorithm for unary outliers (ECDA-UO) for single edge data timing continuity detection. *RelationSpout* will receive the relational model among the different parameters of edge data sent by the user. Relation Spout will send the relational model to the fork of edge computing detection algorithm for multivariate parameter relationship (ECDA-MPR) for detection relationship of edge devices. If the relational model set is relatively large, it can also be considered to use the data partitioning module to divide and resend it to the corresponding ECDA-MPR. After ECDA-UO completes the timing correlation detection, multiple nodes of ECDA-UO will send corresponding edge data to the corresponding ECDA-MPR, and Data relationship check was performed at ECDA-MPR. Meanwhile, ECDA-UO will also send the results of the timing continuity check to Fusion. After waiting for the corresponding relation result in ECDA-MPR to be detected, ECDA-MPR will also send the corresponding correlation detection result to Fusion to complete the final multi-source edge data anomaly detection. The user can also send the corresponding query information to *QuerySpout*, and *QueryPoint* will receive the query information of the user, to query the corresponding data abnormality according to the user's request and output.

4.1 Edge computing detection algorithm for unary outliers

This section will give a definition of the unary and multivariate parameter relationship data outliers detection.

Definition 1. The unary data collected by edge devices and transmitted with the form of time series data, which can be simplified:

$$TS_m=\{S_1, S_2,\ldots, S_i,\ldots, S_m\} \tag{1}$$

$$S_i=\{s_1,s_2,\ldots, s_j,\ldots,s_n\} \tag{2}$$

where: $1\leq i \leq m$, $1\leq j \leq n$, TS_m represents a time series representation data set of multi-source edge data. m represents the number of data in the set. In the formula (1), S_i represents a single edge data, and in the formula (2), n represents the length of S_i. Where s_j represents the data value of a specific acquisition time, $s_j=(v_j, t_j)$, t_j represents the time stamp of s_j, v_j represents the data value of t_j, and t_j is strictly incremented in the time series.

According to the time series representation S_i of the single edge data above, we will introduce a sliding window (slide windows, SW) [46] to store part of the data of S_i, and set the length of the SW to Len_{sw} and ignore the time series data in the SW. For the label, we give the definition of the outlier distribution of the time series data in the SW.

$$\mu = \sum_{i=1}^{n} v_i / n \tag{3}$$

$$\sigma = \sqrt[2]{\sum_{i=1}^{n} (v_i - \mu)^2 / n} \tag{4}$$

Definition 2. Partial time series in SW can be simplified as: $ST_n=\{v_1, v_2,\ldots,v_t,\ldots,v_n\}$ ($1\le t\le n$), then the mean of n points is defined as mean u(3) and variance σ(4), under the assumption of normal distribution, the region $\mu+3\sigma$ contains 99.7% of the data, if v_t ($1\le t\le n$) and the unary outlier distribution of all the data in the SW. If the mean μ exceeds 3σ, then this value can be marked as an outlier.

Based on the definitions, ECDA-UO is described as follows:

Algorithm 1. ECDA-UO

Input: *Time series data TS, length of sliding window SW* Len_{sw}, *subsequence moving distance* Len_{move}, *minimum length threshold of sliding window* ε_{size} *and relative outlier distance threshold* 3σ;

Output: *result of outliers set* Ω_{ab}.

$\Omega ab=\emptyset$; / Initialization parameter set */*
*Hashmap MapForUO=new HashMap();/*A new Hashmap to store exception parameters */*
qTS=InitQueue(Len_{sw}), listSW=InitList(Len_{sw});/ Initialize data queue and sliding window list */*
while TS.length()>Len_{sw}
    caclcSWD is (SW,TS,Len_{sw});/ Output Len_{sw} from TS into SW and calculate Outlier */*
    IF $\mu(v_i)>3\sigma$ && Len>ε_{size}
        qTS.enQueue(ts_{sub});/ Put the subsequence ts_{sub} into the queue qTS */*
    End if
    *while qTS.length()$\neq$0/ Select ts_{sub} from the queue qTS to judge again */*
        tssub=qTS.deQueue();
        if calcValue $\mu(ts_{sub})> 3\sigma$&&ts_{sub}.length<ε_{size}
            *$\Omega ab=\Omega ab \cup ts_{sub}$ /*Put ts_{sub} into Ωts_{sub} */*
        else
            *qTS.enQueue(ts_{sub}); /*Reduce the len_{move} of ts_{sub} again, create a new ts_{sub} */*
        End if
    End while
*mapForUO.put(abID,ts_{sub});/*Put outliers into Hashmap*/*
Return mapForUO. / * End of algorithm */

ECDA-UO mainly uses the time series continuity of the edge data itself and detects the abnormalities that may occur in the edge data by calculating the mean and variance of the relative unary outlier distribution. The algorithm can detect data anomalies of single source edge data.

4.2 Edge computing detection algorithm for multivariate parameter relationship

The edge data acquired usually has a certain relation. One can use the relationship among edge devices to determine whether an edge data has an abnormality.

Definition 3. According to a certain correlation known in the multi-source time series $TS_m=\{S_1, S_2,\ldots, S_m\}$, the necessary combination and transformation of S_m is performed to obtain a time series S'_k satisfying the multivariate linear correlation. And put it into the correlation parameter set Ω_k, denoted as $\Omega_k=\{S'_1, S'_2,\ldots, S'_k\}$.

According to definition 3, we carry out the necessary combination and transformation operations of the partial time series set TS'_{sub} of TS'_m that satisfy the known correlation, making it a multi-source time series TS'_k with linear correlation and put them into different parameter sets Ω_k respectively, and then verify whether the corresponding linear correlation constraints are met between the actual observations of the edge data and Ω_k. We will use the TS_m of SW as the starting point, and the corresponding TS_m correlation detection is performed. However, there may be no corresponding linear correlation or nonlinear correlation in TS_m, so TS_m first needs to be converted into multi-source time series TS'_k with linear correlation. In order to ensure the smooth progress of subsequent tests. Based on the above considerations, ECDA-MPR is described as follows:

Algorithm 2. ECDA-MPR

Input: Exception ID list listForAB, mapForUO, mapForMPR;

Output: Exception result set Ω_{result}.

$\Omega_R=\Omega_E=\Omega m=\emptyset,\Omega_{result}=\emptyset$;/* Initialize the parameter set and exception result set */
$\Omega_{del}=\Omega_{add}=\emptyset$;/* Initialize 2 temporary collections for integration of exception data */
Hashmap mapForResult=new HashMap();/*Built a Haspmap to store exception results*/
Hashmap mapForMPR=new HashMap();/*Built a Haspmap to store exception parameters */
listTS=InitList(TS)
while length(Ω_k)$\neq 0$
  item= Dequeue(Ω_k);
  stp_i=listTS.get(p_i);/* Fetches the corresponding data by the sensing timing p */
  corr=corrDetc(item, stp_i, stp_j...);/* Verify relevant time series data meets constraints */
  If corr is true
    $R=\Omega_R \cup$item;/* Incorporate parameters of related time series data into Ω_R*/
  Else
    $\Omega_E=\Omega_E\cup$ item;
  End if
End while
$\Omega_m=\Omega_E-\Omega_R$;/* Get the exception parameter set */
Ωc=mapForUO.get(abID); /* Get the exception parameter set */
  If $\Omega_m\neq\emptyset$ && $\Omega_c\neq\emptyset$/* Both algorithms are enabled* /
    for each ab_m in Ω_m
      if $ab_m\notin \Omega c$
        Findvalue(ab_m, Ω_m);/*Find $ab_i\notin\{\Omega k\text{-}ab_m\}$*/
        $\Omega_{del}=\Omega d_{del}\cup ab_i$;
      End if
    End for
  for each $ab_c\in\Omega c$ &&$ab_c\in\Omega_R$
    Findvalue (ab_c,Ω_R);/* Find $ab_i \in\Omega_k\text{-}ab_c$*/
  $\Omega_{add}=\Omega_{add}\cup ab_i$;

```
        End for
    Ωresult=Ωc ∪ (Ωm-Ωdel) ∪ Ωadd;
      If Ωresult≠∅/* The loop ends and the exception in Request is stored in Hashmap*/
        for each abi in Ωresult
          mapForResult.put (abID, abi)/* Store abnormal results in Hashmap*/
        End for
      End if
     End if
    End for
    Return mapForResult. / * End of algorithm */
```

ECDA-MPR combines the abnormal result sets of edge data acquired by the timing and correlation algorithms, which mainly optimizes the abnormal detection results of ECDA-UO, supplements the data anomalies that ECDA-UO can't find, and also eliminates the corresponding data without abnormalities. According to the detailed flow of ECDA-MPR, the computational complexity of ECDA-MPR is $O(n^2)$

5. Prototype development

The edge data acquisition device in the prototype system is shown in Figure 5. Accelerometer detached and attached are used with machine tool. The data items that the prototype system can collect include: tool number, tool position, spindle speed, and machine status. Since the limited communication speed between the PC and the controller by Python, the frequency of the collected data is about 1-2 seconds. The tool position and spindle speed collected at low frequencies can still help identify the operation of the machine tool when it is associated with G code. The collected edge data is stored in a local PostgreSQL database. Users can access data by using a valid username and password.

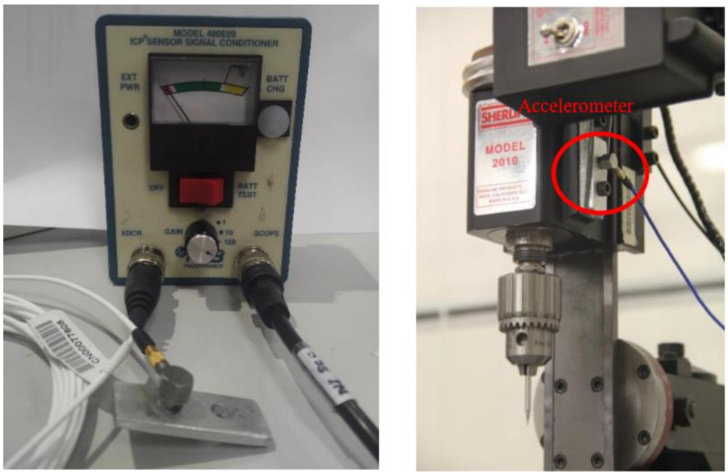

Figure 5. Accelerometer detached (left) and Accelerometer attached (right)

Edge data transmitted by MTConnect format was acquired using a National Instruments PCI-6221 data acquisition card in Figure 6. It is capable of sampling analog and digital signals and is compatible with Linux-based drivers. It also provides the signal conditioner SCC-ACC01 with the ability to amplify the acceleration signal. The actual position, speed and acceleration of each axis of the machine can be obtained. Each axis is connected to an encoder, which is a relative encoder that provides a frequency proportional to its speed. Although this can measure speed of the spindle immediately, the data must be further processed to determine the actual acceleration and jerk. A convenient method is to connect all encoders to the display box, display the actual position of each axis and the spindle speed, and position all machine axes at the zero point of the coordinate system before starting the acquisition. Use the comedic driver collection library to periodically capture data and store it in an internal database. The position sensor is an optical encoder that outputs a sequence of numbers; the order varies depending on the direction of motion. The distance and direction of motion are determined by reading the sequence and parsing it.

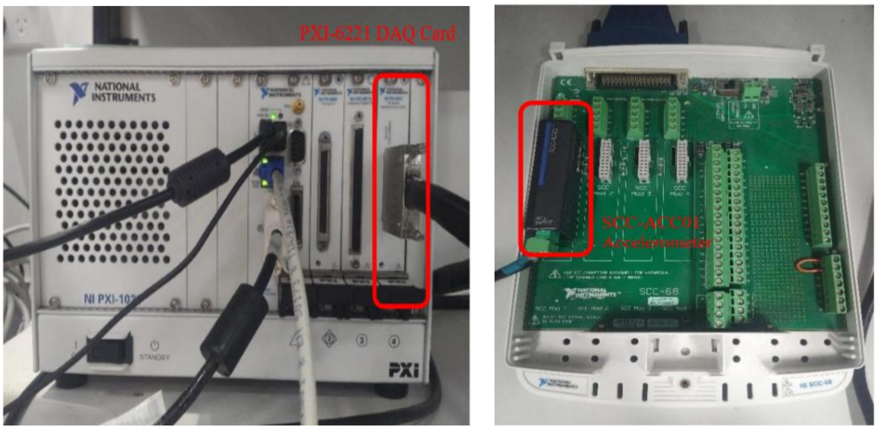

Figure 6: Shield Connector Block SCC-68

5.1 MTConnect agent

The MTConnect agent is the main part of providing the MTConnect interface. It uses HTTP as the agent to process MTConnect requests and respond to the corresponding MTConnect data stream. In the best case, the agent is embedded in the machine controller and sends data directly from within the controller. For machines that cannot be directly supported, MTConnect standard provides a version of the agent, which is divided into two parts: the agent itself and MTConnect adapter. Although the agent still provides MTConnect interface, it no longer provides data directly to the controller. It first gets the data from MTConnect adapter. This separation allows MTConnect agent to maintain a wide range of versatility, while the adapter can be highly customized to meet the controller's requirements. In an agent, data samples for its devices are stored in buffers of configurable length. Each new sample stored in this buffer is marked with a millisecond-precise timestamp and a unique incremental sequence ID to identify the original order of the data samples.

5.2 MTConnect adapter

The Adapter is a software application which gathers data from a device and streams this data to the agent in a standard format [37]. If an agent is used and is detached from the adapter, data collection will first be sent to MTConnect adapter. Although the adapter and the agent communicate via a standard port such as TCP/IP, the adapter can be directly connected to the machine controller or sensor. It's easy to establish a connection to a dedicated hardware platform.

As a connection between the machine tool and MTConnect agent, the task of the adapter is to collect data from two different sources. The first is a data collection agent that provides its data through a TCP agent. The second source is the EMC2 controller, which provides internal status such as processing mode, command location or other path related information. To get this data, the adapter writes an EMC2 controller-based plug-in in C++. The ability to transfer additional data values from a data acquisition agent is achieved by extending the adapter functionality. The collected data is then packaged into a time-stamped string and passed to the agent by a TCP connection.

DAQ Assistant is used to receive the raw data from the adapter in Figure 7. Testing is conducted by switching on the Accelerometer and pressing the RUN button on the DAQ Assistant interface. Once there are waves (small fluctuation) occur as shown on the left below, it means the accelerometer is connected.

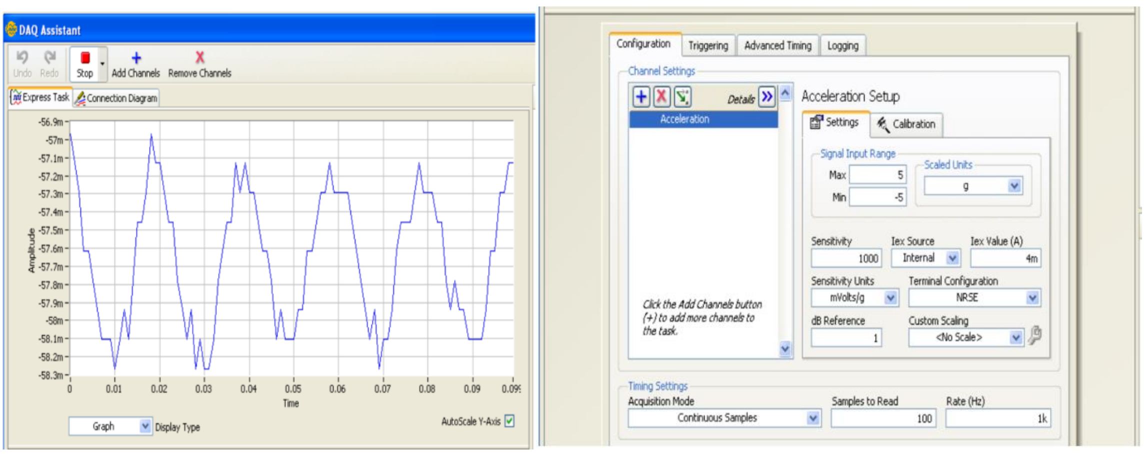

Figure 7. DAQ Assistant from adapter.

5.3 DT in the local

As shown in Figure 8, the client is a DT model image interface using MTConnect standard. In the right area of the main window is the data item selected for visualization in the device browser. The data item includes the actual value and other information (e.g., timestamp, data item unit or subtype) displayed in the upper area. If the data item type is "sample", the past values are also plotted over time. To resolve differences from MTConnect or the underlying protocol, each level of MTConnect is as transparent as possible.

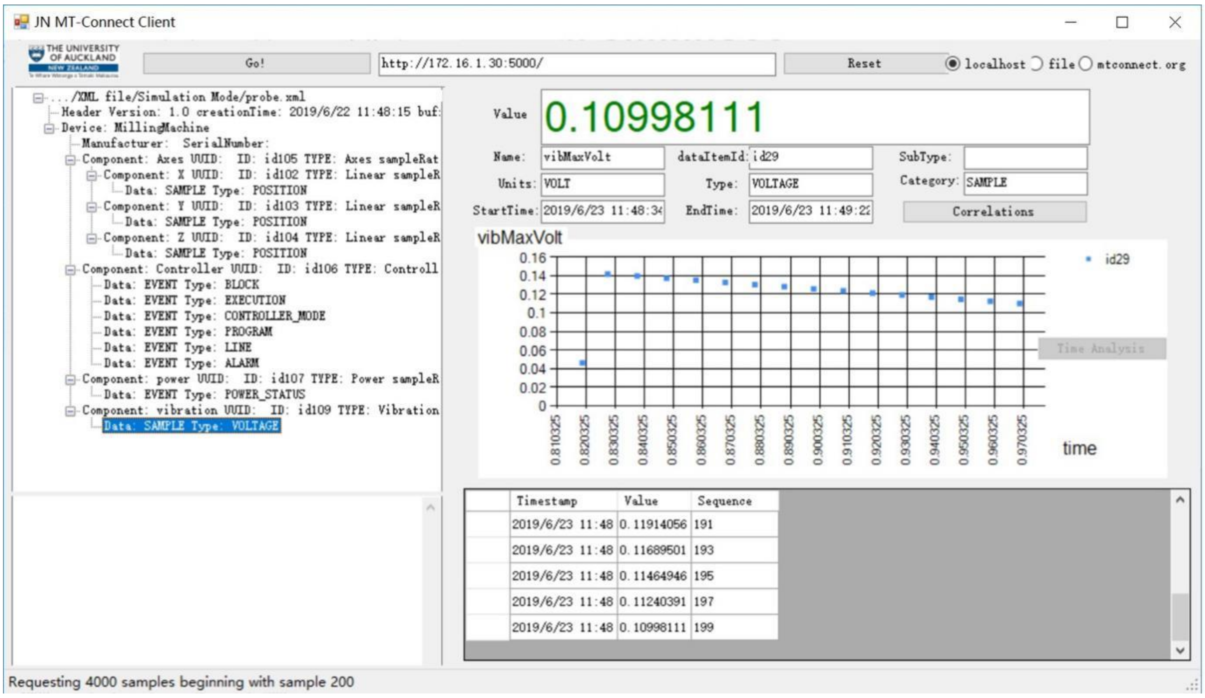

Figure 8. Local data driven CPMT in production with MTConnect.

To realize the experimental analysis of the whole process of pre-production, mid-production and post-production of the processing process, and construct a full mapping of CPMT. Based on this requirement, the implementation of the client will be implemented using Microsoft .NET and Visual Studio 2018. The drawing area is implemented using Visual Studio's Microsoft Chart Control extension. It provides a way to draw different types of graphs. Once the control element is embedded in the GUI, its appearance can be configured through the Visual Studio IDE. Data visualization is achieved through the method of drawing data provided by the runtime. The first graphic of the drawing area is bound to the list of values passed. These lists of values are obtained directly from the DataSequence object stored in the Device structure by reading the getTimeScaleAsList of X-axis and the getSampleValueAsList of Y-axis. Even though these lists contain many elements, the entire read process is still very lazy with no delay and can be refreshed in real time as each change in the data. The 3D animation of CNC machine tools is not discussed here. This paper only focuses on the DT model and implementation logic of CPMT.

5.4 DT in the cloud

Through the analysis of real-time data, the invisible processing process is made explicit. As shown in Figure 9, the client of cloud-driven CPMT in production with MTConnect is displayed in real time.

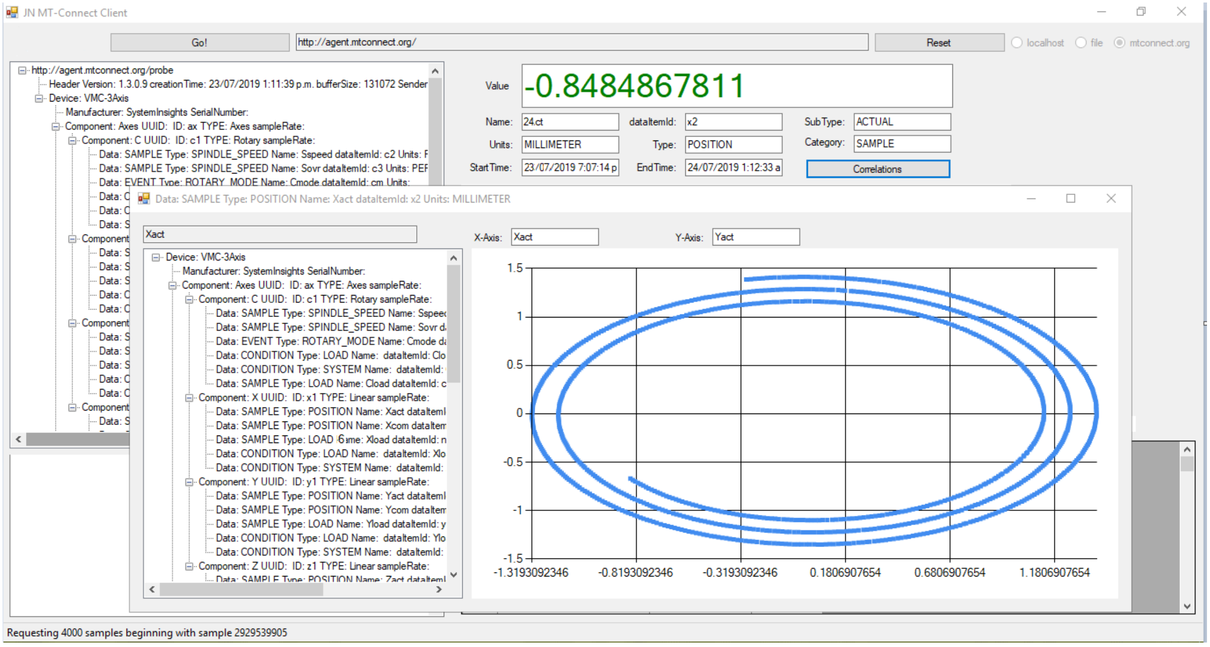

Figure 9. Cloud-driven CPMT in production with MTConnect.

DT data includes the physical data and the virtual data: 1) the physical data collected from sensors in the workshop and machine tool, 2) the virtual data came from the virtual models and production systems (i.e. ERP, MES). The cloud-driven CPMT in production can optimize three functions including resource management, process control and production planning.

Firstly, in terms of resource management, raw materials and processing equipment should be allocated according to the production tasks of components. Virtual data from the steel bars mechanical/thermal analysis data, and running failure data can be obtained from virtual models of steel bars and machine tool. Based on the above data, and the data processed by association, clustering, regression, etc., the service from the workshop can design a plan for distributing steel bars and machine tool for the current processing task.

Secondly, the plan is transmitted to virtual machine tool for verification before the actual execution. With Cloud-driven CPMT in production in Figure 9, existing problems of machine tool can be found including collision and friction between the tool and the workpiece. Meanwhile, the simulation is repeated at a small cost, iterative testing can be used to optimize the machining plan to achieve higher machining accuracy.

Thirdly, the machine tool starts working with the machining plan. The position of tool, spindle speed, feed rate, etc. can be obtained from CNC system in real-time. Virtual machine tool can update its status based on those data. Meanwhile, the virtual models compared with the processing status, if there is an inconsistency in the results, services of CPMT will evaluate the process to determine whether the results are caused by physical interference.

According to the results, a virtual NC machine tools will generate real-time command to standardize the processing or change the machining plan. After the manufacturing process is completed, Dimensions, accuracy, balance and other indicators need to be tested. If the indicators in the virtual product meet the requirements, the processed product is qualified, otherwise repairs are required.

6. Algorithm Experiment

6.1 Test verification set

To verify the anomaly detection capability of edge computing algorithm, the interpolation process of machine tool is used as the test verification set. According to the different types of acceleration and deceleration, the speed control can be divided into linear, trigonometric, exponential acceleration and deceleration, S-curve and quadratic curve acceleration and deceleration. In general, there is a derivative relationship between displacement curve, velocity curve, acceleration curve and jerk curve. Taking the quadruple curve acceleration and deceleration as an example:

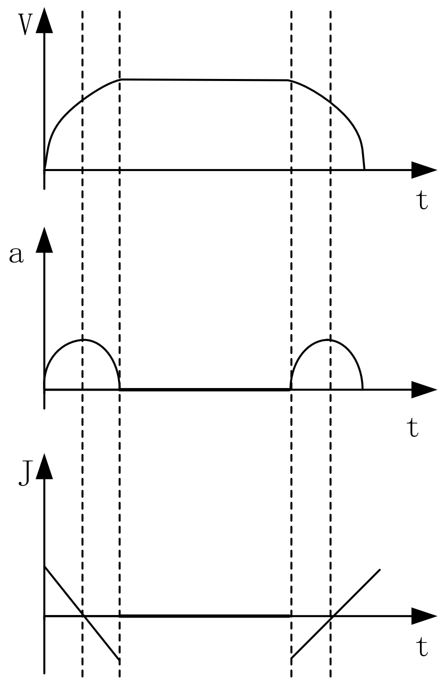

Figure 10. Schematic diagram of velocity, acceleration and jerk based on four displacement curves.

6.1.1 Verification data set construction

The data set consists of machine speed, acceleration, jerk, etc., D(u) represents displacement, and a_0~a_4 represents coefficient:

$$D(\mathrm{u}) = a_0 + a_1 u + a_2 u^2 + a_3 u^3 + a_4 u^4 \tag{5}$$

From the above reciprocal relationship, the velocity V, the acceleration a, and the jerk J are respectively taken as:

$$\begin{cases} V(u) = a_1 + 2a_2\mathrm{u} + 3a_3 u^2 + 4a_4 u^3 \\ a(u) = 2a_2\mathrm{u} + 6a_3\mathrm{u} + 12a_4 u^2 \\ J(u) = 6a_3 + 24a_4\mathrm{u} \end{cases} \tag{6}$$

where $u = t/\mathrm{t_m}$: $\mathrm{t_m}$ is the time of acceleration or deceleration process, t is the time of acceleration or deceleration, t∈[0, $\mathrm{t_m}$]. The following boundary conditions must be met at the start and end:

$$\begin{cases} D(0) = 0 \\ V(0) = V_1 \\ V(1) = V_2 \\ a(0) = 0 \\ a(1) = 0 \end{cases} \tag{7}$$

Among them, V_1 and V_2 are the starting velocity and the ending velocity of the machining track segment. By the boundary condition (6), the displacement, velocity, acceleration and jerk curves (7), the coefficients a_0~a_4 can be calculated and brought into the curve equation(8):

$$\begin{cases} D(t) = V_1\mathrm{t} + \frac{V_2-V_1}{t_m^3}t^3 + \frac{V_1-V_2}{2t_m^3}t^4 \\ V(t) = V_1\mathrm{t} + \frac{3(V_2-V_1)}{t_m^2}t^2 + \frac{2(V_1-V_{2)}}{t_m^3}t^3 \\ a(t) = \frac{6(V_2-V_1)}{t_m^2}\left[t - \frac{t^2}{t_m}\right] \\ J(t) = \frac{6(V_2-V_1)}{t_m^2}\left[1 - \frac{2\mathrm{t}}{t_m}\right] \end{cases} \quad (8)$$

6.1.2 Test case

A total of 100,000 pieces of sensor data in the data set are selected, and the actual observed values of the sensor data are shown in Table 1. According to the corresponding acceleration principle makes it easy to find that the displacements *D*(t) and $\Delta V/\Delta t$ and a(t) and $\Delta V/\Delta t$ have nonlinear correlations.

Then we use the formula of the correlation coefficient to calculate the velocity data set accordingly. The algorithm proposed in this paper is used to verify the anomaly data. According to the detection results, there are 460 *V*, $\Delta V/\Delta t$, a(t) anomaly data in 100000 edge data. The total number of abnormal data can be expressed as AB_{sum}, and the successfully detected abnormal data can be expressed as AB_{cor}, and the detection accuracy of the abnormal data AB_{ac} is calculated as:

$$AB_{\mathrm{ac}} = AB_{\mathrm{cor}} / AB_{\mathrm{sum}} \quad (9)$$

Table 1. Velocity data statistics

Linear acceleration		S-curve acceleration		Quadratic curve acceleration	
Time（s）	Velocity（mm/s）	Time（s）	Velocity（mm/s）	Time（s）	Velocity（mm/s）
0~0.08	3.25~26	0~0.14	1-25	0~0.34	1.08-25
0.08~0.4	24	0.14~0.4	25	0.34~0.42	25
0.4~0.48	24~1	0.4~0.58	25~1.06	0.42~0.68	25~1.36
0.48~0.6	1~24	0.58~0.66	1.06~19.5	0.68~0.86	1.36~14.5
0.6~0.72	24~2.82	0.66~0.82	19.5~1.06	0.86~0.93	14.5~7.06
0.72~0.8	2.82~24	0.82~0.94	1.06~26	0.93~1.24	7.06~26
0.8~1.72	24	0.94~2	26	1.24~2.25	26

1.72~2	24~2.63	2~2.24	26~1.08	2.25~2.44	26~4.28
2~2.26	2.63~24	2.24~2.36	1.08~26	2.44~2.56	4.28~18.9
2.26~2.32	24	2.36~2.32	26	2.56~2.78	18.9~1.08
2.32.6~2.46	24~1	2.3~2.59	26~1.08	-	-

6.2 Analysis of abnormal detection results

6.2.1 Data abnormal detection for ECDA-UO and ECDA-MPR

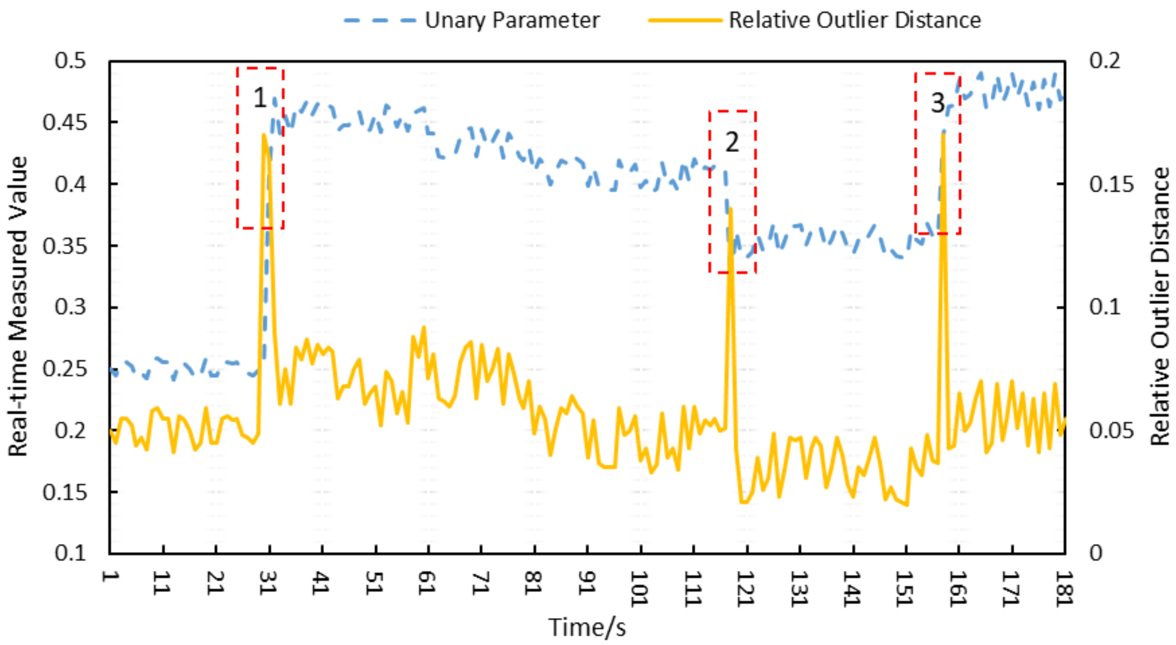

Figure 11. Data abnormal detection for ECDA-UO by unary parameter

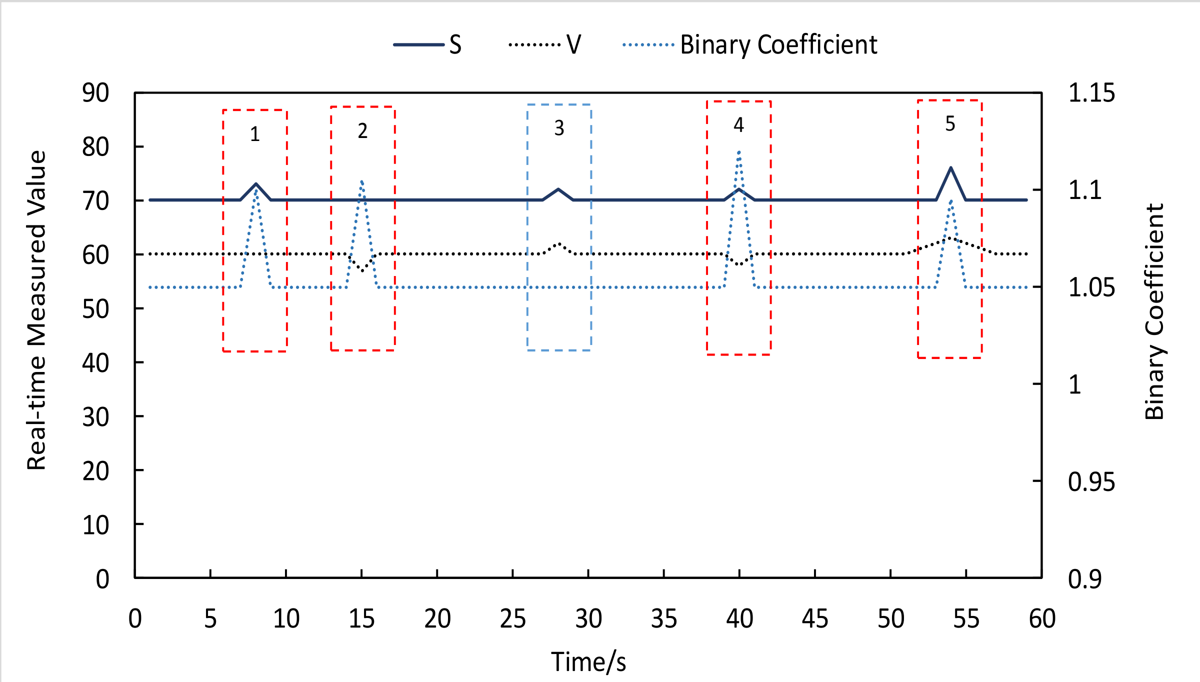

Figure 12. Data abnormal detection for ECDA-UO by multivariate parameter

ECDA-UO uses the temporal continuity of the edge data itself and calculates the relative outlier distance, which can detect data anomalies of single-source edge data. As shown in Figure 11, with unary parameter, the data anomalies of the edge data in boxes 1, 2 and 3 can be detected. However, ECDA-UO only considers the inherent time series continuity of single-source edge device data, and ignores the correlation between multi-source edge data. Therefore, there may be a problem that some data anomalies cannot be detected effectively by unary parameter.

ECDA-MPR mainly uses the multivariate parameter among the sensing data (lines 1-28 of algorithm 2) to detect possible abnormalities in the edging data. If ECDA-MPR only considers the relationship among edge devices, and ignores the time series continuity of edge data itself. There may be two problems:

1) If there are fewer elements in the parameter set Ω_k, it is difficult to accurately locate the abnormal edge data by the relationship among edge devices. Consider a set of linear relationship sequences $TS_2 = \{S_1, V_2\}$ in the parameter set Ω_2. After the relationship test of multivariate parameter, we found that there are abnormal data in the sensor data (S_1, V_2). As shown box 1, box 2, box 4 and box 5 in Figure 12, there are data abnormalities in (S_1, V_2), whether S_1 or V_2 is abnormal, or both of them are abnormal, it is difficult to locate.

2) If all parameters in the parameter set Ω_k are abnormal at the same time, the corresponding data abnormality may not be successfully detected by the multivariate parameter. As shown box 3 in Figure 12, when $TS_2 = \{S_1, V_2\}$, S_1 and V_2 have data abnormalities at the same time, and the abnormal data also meet the constraints of the corresponding binary linear model, then the corresponding abnormal data cannot be detected by multivariate parameter.

Therefore, ECDA-MPR is proposed based on ECDA-UO, which adopts an effective data fusion process (FP) on the two test results (line 29-37 of algorithm 2) to obtain more accuracy abnormal data.

6.2.2 Comparison of abnormal detection accuracy

ECDA-UO can only find 334 out of 460 abnormal data in the quad curve acceleration and deceleration data set (*AB*ac=0.73). ECDA-MRP can use the multivariate parameter relationship to find 420 out of 460 abnormal data (*AB*ac=0.91), and can accurately locate the abnormal data. ECDA-MRP algorithm proposed in this paper can perform data fusion operations on the unary and multivariate parameter relationship data outliers. Therefore, the detection results of ECDA-MRP are significantly better than ECDA-UO. Based on all the edge data in the dataset, ECDA-UO, ECDA-MRP and benchmark methods (AD_IP [47], AD_KNN [36]) are used to detect the abnormality of the sensor data, and experimental results are compared and analyzed in Figure 13. The detection result of ECDA-MRP is significantly better than ECDA-UO. Although the two comparison benchmark methods are based on time series important point segmentation and k-nearest neighbor search based on fast selection strategy to find corresponding anomaly data, the above methods do not make good use of the "widespread" correlation between multivariate parameter relationships, so that multi-source related data anomalies cannot be effectively identified. Therefore, ECDA-MRP algorithm not only has strong anomaly detection capability, but also can reduce the pressure on the cloud service center.

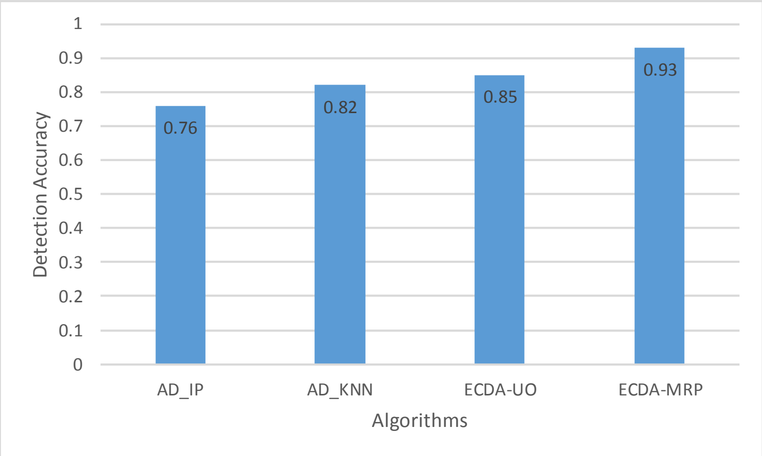

Figure 13. Comparison of abnormal detection accuracy.

6.3 Time-Consuming in edge computing and cloud computing

In the data collection process, each machine's sensor aggregates the collected data in the machine and then transmits the data to the edge node. The edge computing test was performed using the data set of the previous section. When the data received by an edge node reaches the threshold, or the waiting time reaches the threshold, the currently received data is all transferred to default DataSpout port and distributes the relevant data to ECDA-UO to start timing continuity detection. After ECDA-UO completes the timing correlation detection, ECDA-UO will send the corresponding edge data to the corresponding ECDA-MPR for data correlation check. At the same time, ECDA-UO will send the timing continuity check result to Fusion. ECDA-MPR will also send the corresponding correlation detection result to Fusion.

Table 2. Comparison of Time-Consuming in edge computing and cloud computing

Operation	Time-Consuming in Edge Computing/ms	Time-Consuming in Cloud Computing/ms
Receive	256.3	342.1
Save	156.3	281.6
ECDA-UO	2.3	2.8
ECDA-MPR	0.76	0.9

According to different processing methods, the corresponding processing average time is as shown in Table 2. The cloud computing receiving data is larger in scale and the bandwidth pressure is higher. The network transmission time is significantly longer than the edge computing. Because the computing power of the cloud center node is relatively strong, the time spent on the anomaly detection is shorter than the edge calculation. Due to the pressure of data size and bandwidth, the processing time of cloud computing is relatively high. Therefore, the detection anomaly detection performance of ECDA-MPR is better.

7. Conclusion and future work

Machine tool is transforming from isolated manufacturing unit to intelligent service providers. By DT and edge computing technique to build a CPMT, the virtual and real mapping can be realized, and the machining process of the physical machine tool can be guided according to the simulation result. A local and cloud DT application are developed at the application layer. In order to eliminate the time delay and low precision of cloud modeling, a novel edge computing algorithm is proposed to detect the abnormality of the edge data from two aspects: the unary outliers of the edge data itself and the multivariate parameter correlation among edge devices.

This work still has some limitations. One of statements made by the authors is that, owing to commercial privacy restrictions, incompatibility between machine tools and CNC system interfaces, edge data detection methods and digital twin models are not applicable to all machine tools. Therefore, digital twin application needs to be made based on currently open protocols and interfaces. Nevertheless, this work presents a case study of an edge computing-based cyber-physical machine tool and more open discussions are welcome in this research area.

Meanwhile, a few potential research points are indicated, to be specific: (1) Enrich the data visualization and data analysis such as a bar diagram, curve diagram or 3D diagram depend on the data that need to be displayed. The mature 3D modeling is introduced into the software to realize the 1:1 modeling of physical machines and virtual machines. (2) Cloud database. Currently the data values are stored in a local file instead of a cloud database. Future work can consider how to upload and download data from the cloud. (3) The operating mechanism of CPMT in the CPPS. One of the values of CPMT is how to combine production systems to improve efficiency. New requirements for DT need to be explored. For example, DT application can endow the physical objects with a certain degree of autonomy.

Acknowledgements

This work was supported by the National Science and Technology Major Project of the Ministry of Science and Technology of China (Grant No. 2019ZX04014001-004). The authors would like to thank Dr. Chao Liu from the University of Auckland for supporting the experimental work. The authors are also grateful to the researchers at the Laboratory for Industry 4.0 Smart Manufacturing Systems, the University of Auckland. Finally, we wish to thank the editor and the anonymous reviewers for helping to strengthen this paper.

References

[1] X. Xu, Machine Tool 4.0 for the new era of manufacturing. The International Journal of Advanced Manufacturing Technology, 2017, 92 (5-8): 1893–1900.

[2] X. Xu, From cloud computing to cloud manufacturing. Robotics & Computer Integrated Manufacturing, 2012, 28(1):75-86.

[3] F. Tao, L. Zhang, and Y. J. Laili. CLPS-GA for energy-aware cloud service scheduling, in Configurable Intelligent Optimization Algorithm: Design and Practice in Manufacturing. New York, NY, USA: Springer, 2014, pp. 191–222.

[4] F. Tao, M. Zhang. Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing. IEEE Access, 2017, 5:20418-20427.

[5] C. Deng, R. Guo, C. Liu, et al. Data cleansing for energy-saving: a case of Cyber-Physical Machine Tools health monitoring system. International Journal of Production Research, 2017(2):1-16.

[6] L. Catarinucci, D. D. Donno, L. Mainetti, et al. An IoT-Aware Architecture for Smart Healthcare Systems. IEEE Internet of Things Journal, 2015 2 (6): 515–526

[7] P. Zheng, T. J. Lin, C. H. Chen, X. Xu, A systematic design approach for service innovation of smart product-service systems. Journal of Cleaner Production, 2018, 201: 657-667.

[8] B. R. Seshadri, T. Krishnamurthy. Structural Health Management of Damaged Aircraft Structures Using the Digital Twin Concept, 25th AIAA/AHS Adaptive Structures Conference. 2017: 1675. doi: 10.2514/6.2017-1675.

[9] K. Y. H. Lim, P. Zheng, C.-H. Chen. Journal of Intelligent Manufacturing, 2019, 1-25. https://doi.org/10.1007/s10845-019-01512-w.

[10] Digital twin | Siemens, (n.d.). https://www.plm.automation.siemens.com/global/en/our-story/glossary/digital-twin/24465 (accessed April 24, 2019).

[11] P. Zheng, Z. X. Wang, C.-H. Chen, L. P Khoo. A Survey of Smart Product-Service Systems: Key Aspects, Challenges, and Future Perspective. Advanced Engineering Informatics, 2019, 42: 100973. https://doi.org/10.1016/j.aei.2019.100973.

[12] W. Luo, T. Hu, C. Zhang, Y. Wei, Digital twin for CNC machine tool: modeling and using strategy. Journal of Ambient Intelligence and Humanized Computing, 2019, 10(3): 1129–1140.

[13] E. H. Glaessgen, D. Stargel. The Digital Twin Paradigm for Future NASA and US Air Force Vehicles. 53rd Struct.Dyn. Mater. Conf. Special Session: Digital Twin, Honolulu, HI, US, 2012, 1–14.

[14] M. Grieves, J. Vickers. Digital twin: mitigating unpredictable, undesirable emergent behavior in complex systems. In: Kahlen FJ, Flumerfelt S, Alves A (eds) Transdisciplinary perspectives on complex systems. Springer, Cham. 2017, https ://doi.org/10.1007/9783-319-38756 -7_4

[15] E.J. Tuegel, A.R. Ingraff ea, T.G. Eason, S.M. Spottswood, Reengineering aircraft structural life prediction using a digital twin, International Journal of Aerospace Engineering. 2011, 1–14, https://doi.org/10.1155/2011/154798.

[16] E.M. Kraft. The air force digital thread/digital twin—life cycle integration and use of computational and experimental knowledge. In: AIAA aerospace sciences meeting, 2015, https://doi.org/10.2514/6.2016-0897

[17] G. Schroeder, C. Steinmetz, C. Pereir , D. Espindola, Digital twin data modeling with automationML and a communication methodology for data exchange. IFAC Pap OnLine, 2016, 49(30):12–17. https://doi.org/10.1016/j.ifacol.2016.11.115

[18] F. Tao, M. Zhang, J. Cheng, Q. Q . Digital Twin Workshop: A New Paradigm for Future Workshop. Computer Integrated Manufacturing Systems, 2017, 23 (1): 1–9

[19] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui. Digital Twin-driven Product Design, Manufacturing and Service with Big Data. International Journal of Advanced Manufacturing Technology, 2018, 94, (9–12):3563–3576.

[20] A. Moreno, G. Velez, A. Ardanza, et al. Virtualization process of a sheet metal punching machine within the Industry 4.0 vision. Int J Interact Des Manuf (IJIDeM), 2017, 11(2):365–373. https://doi.org/10.1007/s12008-016-0319-2

[21] R. Magargle, L. Johnson, P. Mandloi, P. Davoudabadi, et al. . A simulationbased digital twin for model-driven health monitoring and predictive maintenance of an automotive braking system. In: The international modelica conference, Prague, Czech Republic, 2017. https://doi.org/10.3384/ecp1713235

[22] Liu B. F., Zhang Y. F., Zhang G., Zheng P., Edge-cloud orchestration driven industrial smart product-service systems solution design based on CPS and IIoT. Advanced Engineering Informatics, 2019, 42: 100984. https://doi.org/10.1016/j.aei.2019.100984

[23] A. Sharma, H. Chen, D. Min, et al. Fault detection and localization in distributed systems using invariant relationships, IEEE/IFIP International Conference on Dependable Systems & Networks. 2013.

[24] M. Han, L. Jin, A. Kang, et al. A Statistical-Based Anomaly Detection Method for Connected Cars in Internet of Things Environment, Internet of Vehicles - Safe and Intelligent Mobility. 2015..

[25] G. Liao, J. Li. Distance-Based Outlier Detection for Distributed RFID Data Streams[J]. Journal of Computer Research & Development, 2010, 47(5):930-939.

[26] S. Xia, Z. Xiong, H. Yun, et al. Relative density-based classification noise detection, Optik - International Journal for Light and Electron Optics, 2014, 125(22):6829-6834.

[27] Z. Guo, W. Wong, M. Li. Sparsely connected neural network-based time series forecasting, Information Sciences, 2012, 193(11):54–71.

[28] J. Mourão-Miranda, D. Hardoon, T. Hahn, et al. Patient classification as an outlier detection problem: An application of the One-Class Support Vector Machine, Neuroimage, 2011, 58(3):793-804.

[29] J. Wang, J. Chiang. A Cluster Validity Measure With Outlier Detection for Support Vector Clustering, IEEE Transactions on Systems Man & Cybernetics Part B

Cybernetics A Publication of the IEEE Systems Man & Cybernetics Society, 2008, 38(1):78-89.

[30] T. Fu. A review on time series data mining, Engineering Applications of Artificial Intelligence, 2011, 24(1):164-181.

[31] M. Vlachos, P. Yu, V. Castelli, et al. Structural Periodic Measures for Time-Series Data, Data Mining & Knowledge Discovery, 2006, 12(1):1-28.

[32] Cattivelli, Federico & H. Sayed, Ali. Distributed Detection Over Adaptive Networks Using Diffusion Adaptation. Signal Processing, IEEE Transactions on. (2011) 59. 1917 - 1932. 10.1109/TSP.2011.2107902.

[33] A. Anandkumar, L. Tong, and A. Swami. Detection of Gauss-Markov Random Fields With Nearest-Neighbor Dependency. IEEE Trans. Inf. Theory, 55(2):816－827, Feb 2009.

[34] A. Anandkumar, Scalable algorithms for distributed statistical inference. 2009

[35] R. Fujimaki, T. Yairi, K. Machida, An Anomaly Detection Method for Spacecraft Using Relevance Vector Learning. 2005.

[36] L. Cai, N. Thornhill, S. Kuenzel, et al. Real-time Detection of Power System Disturbances Based on k-Nearest Neighbor Analysis. IEEE Access, 2017, PP (99):1-1.

[37] Liu C, Vengayil H , Lu Y , et al. A Cyber-Physical Machine Tools Platform using OPC UA and MTConnect. Journal of Manufacturing Systems, 2019, 51:61-74.

[38] Zhu Z, Liu C, X. Xu, Visualization of the Digital Twin data in manufacturing by using Augmented Reality. Procedia CIRP 81 (2019) 898–903.[37] MTConnect Institute, Standard—MTConnect, (n.d.). https://www.mtconnect.org/ standard 20181(accessed April 24, 2019).

[39] ISO, ISO 14649-1: iIndustrial automation systems and integration - Physical device control - Data data model for computerized numerical controllers - Part 1: overview and fundamental principles, (2003).

[40] T. Skeie, S. Johannessen, O. Holmeide, Timeliness of Real-Time IP Communication in Switched Industrial Ethernet Networks, IEEE Transactions on Industrial Informatics, 2006, 2(1):25-39.

[41] C. Deng, R. Guo, X. Xu, R. Zhong, & Z. Yin, A new high-performance open CNC system and its energy-aware scheduling algorithm. International Journal of Advanced Manufacturing Technology, (2017) 93 (5-8), 1513-1525.

[42] L. Chao, X. Xu, Q. Peng, et al. MTConnect-based Cyber-Physical Machine Tool: a case study. Procedia Cirp, 2018, 72:492-497.

[43] Y. Altintas, C. Brecher, M. Weck, S. Witt, Virtual machine tool. CIRP Annals, 2005, 54(2), 115-138.

[44] Y. Yao, M. Liu, J. Du, L. Zhou, Design of a machine tool control system for function reconfiguration and reuse in network environment. Robotics and Computer-Integrated Manufacturing, 2019, 56(1): 117-126.

[45] P. Zheng, Y. Lin, C. H. Chen, X. Xu, Smart, connected open architecture product: an IT-driven cocreation paradigm with lifecycle personalization concerns. International Journal of Production Research, 2019, 57(8): 2571-2584.

[46] R. Lei, Y. Wei, C. Jin, et al. A sliding window-based multi-stage clustering and probabilistic forecasting approach for large multivariate time series data, Journal of Statistical Computation & Simulation, 2017, 87(13):1-15.

[47] D. Zhou, Y. Liu, W. Ma, Time series anomaly Detection. Journal of Computer Engineering and Applications, 2008, 44 (35): 145-147.