Journal of Manufacturing Systems 65 (2022) 59–69

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Context-aware manufacturing system design using machine learning

Yingxin Ye [a], Tianliang Hu [b,*], Aydin Nassehi [c], Shuai Ji [b], Hepeng Ni [a]

[a] School of Electromechanical Engineering, Shandong Jianzhu University, Jinan, China
[b] School of Mechanical Engineering, Shandong University, Jinan, China
[c] Department of Mechanical Engineering, University of Bristol, Bristol, UK

ARTICLE INFO

Keywords:
Context-aware
Machine learning
Manufacturing system

ABSTRACT

With the development of computer, automation and information technology, workers have more challenges to take care of several devices at the same time. Under this situation, context-aware manufacturing system is proposed to help users capture the most relevant information and make the decision timely. Due to the increased demand for small-batch customized products, manufacturing resources and products frequently change, and this leads to variation of context in manufacturing. Traditional rule-based context-aware manufacturing systems need their rules to be modified manually, which is time-consuming and error-prone under the current variability of the market. To create a framework for updating the context-aware logic automatically, this paper presents a novel notion of applying machine learning techniques in the context-aware manufacturing system design. For the proposed context-aware manufacturing system, components comprising a context model for the manufacturing domain, a machine learning based calibration framework and a context extraction module are designed to improve the update efficiency with less costs. Finally, a test manufacturing scenario is simulated to verify the feasibility of applying machine learning algorithms in context awareness.

1. Introduction

The Industrial 4.0 has raised upsurge of smart manufacturing system. Different manufacturing systems has their own design principle and application emphasis, such as the modular design method for a reconfigurable manufacturing system [1], the multi-objective optimization design method for resilience-based system [2], the simulation-based design method for automated high-mix low volume manufacturing system [3], and the digital-twin based design method for remote semi-physical manufacturing system [4]. While this paper proposed a machine-learning design method for context-aware manufacturing system.

The development of computers, automation and information technologies makes running of devices depend less on human participation, thus improving work efficiency and productivity of each human being. In the resulting system, a human operator has more challenges in dealing with emergencies like failures or safety problems as they are no longer only responsible for a single machine but look after several devices at the same time. Extracting the helpful information from the large volume of context data (including data from installed sensors and NC controllers) and responding quickly to them thus becomes a critical challenge in the emerging intelligent manufacturing industry.

In this situation, the concept of context awareness which aims to provide relevant information and/or services to users based on context [5] warrants attention. A well-known definition of context is made by Dey [6]: “context is any information, which can be used to characterize the situation of an entity”. In manufacturing, an entity may be a location, a person, a machine or a product, and all information that related to their status can be considered as part of context.

In order to apply context awareness in manufacturing to support users, four fundamental questions need to be answered:

- What is the information that needs to be provided?
- When should the information be provided?
- Who accepts the information?
- How is the information provided for users?

Traditionally, these questions are answered by defining rules [7]. A rule for quality control can, for example, be made based on experience that, when the vibration of cutting tool reaches a threshold, show the location of machine tool to an operator on his mobile phone to stop processing, because abnormal vibration may lead to a bad quality

* Corresponding author.
E-mail address: tlhu@sdu.edu.cn (T. Hu).

https://doi.org/10.1016/j.jmsy.2022.08.012
Received 26 March 2022; Received in revised form 2 June 2022; Accepted 29 August 2022
Available online 5 September 2022

product. However, there is a drawback for rule-based context awareness that the definition of trigger context is too rigid and not adaptable for actual context change. For example, setting aside the accuracy of estimated vibration threshold that leads to bad product quality, its value can still vary for many different reasons like the workpiece material or the natural frequency of the machine tools. That means, under current volatile markets where material may be sourced from different providers and various machines can be used in the course of production, such rules would need to be modified by experts very frequently, a process that is time-consuming and error-prone.

The relatively recent concepts of Internet of Thing (IoT) [8] and cloud manufacturing bring advanced sensing, communication and data storage technologies into the industrial field and contribute to the data acquisition and accumulation in the manufacturing process. However, as Singh et al. pointed out, data-rich cannot guarantee information-rich [9]. Hence an approach is required for information mining in adaptive context-aware system design; and, machine learning has been proposed as a potential approach for this [10,11]. These two literatures show context data could go as inputs of different machine learning algorithms, but manufacturing context has its own characteristic, which should be refined for subsequent application. Therefore, this paper designs the context model in workshop firstly to guild the collection and access of manufacturing context data. Then, a common machine learning algorithm selection workflow is proposed with consideration of actual manufacturing context-aware targets, which makes the designed manufacturing system adapt to context change with only a short delay.

In the following sections, this paper reports the results of application of machine learning techniques to build a context-aware manufacturing system and uses a case to illustrate the general machine learning algorithm selection workflow for the development of the context-aware manufacturing system. Section 2 provides a review of the literature in context-aware manufacturing systems design and the application of machine learning algorithms in related manufacturing fields. In Section 3, the framework of proposed context-aware manufacturing system is introduced with derailed specification of the two most important components: the manufacturing context model and the machine learning based context extraction module. Section 4 uses a case study to verify the feasibility of utilizing machine leaning algorithms in a context-aware manufacturing system and compares five machine learning algorithms' performances for the presented case to demonstrate their effectiveness. Conclusions and further research are in Section 5.

2. Literature review

This section reviews literatures from two aspects. Firstly, current existing context-aware manufacturing systems are introduced and the points to be improved under current changeable manufacturing environment are analyzed. Then, combined with available techniques and resources, machine learning algorithm seems a feasible way to improve the context-aware manufacturing system. Therefore, literatures about combining machine learning algorithms and historical data in the industrial filed have been reviewed to show its adaptability for context-aware manufacturing system.

2.1. Context-aware manufacturing systems

Currently, most context-aware applications in manufacturing focus on generating timely actionable intelligence for workers on factory floor and supervisors to respond, which is also one of the primary tasks in smart manufacturing [12]. Stöttinger [7] applied context awareness in mobile safety system(MOSES), which aimed to create a safe working environment for plant maintenance to do work clearance. Ali et al. developed a system that helped users to do order picking in a warehouse more easily based on the information of picking tips and picking checks, which came from sensors on shelves, scales and wearable computers [13]. Ciccarelli et al. presents a tool that supports the monitoring of operators' activities, and implements corrective actions based on the data analysis to make workplace socially sustainable[14].

Context modeling and the structure design of the context-aware system also attract attentions. With the development of the Semantic Web[15] and its effective application in smart manufacturing [16], researchers started importing semantics and ontologies into context models to make them more understandable, both for human and machines. Hynes et al. incorporated ambient intelligence and semantic web technology to help users optimize manufacturing processes by providing decision support [17]. The proposed semantic, ambient, generic and extensible framework (SAGE) used web service modelling language to describe services semantically. Chen, Z., et al. proposed a context-aware manufacturing execution system (MES) with a background of dynamic and complex production process in the workshop [18]. This system made services more attentive, responsive and predictive through sensing and automatically deriving employee's needs from the workshop context. Eirinakis et al. presents a situation-aware manufacturing system framework to identify and predict the disruptions occurred in manufacturing [19]. Ontology, as a knowledge description technology is an enabler for the semantic web and is used for context modelling. In most implementations of ontology based industrial systems, reasoning is realized through business rules, which are recorded by XML as an IF-ELSE flow. Uddin et al. presented an ontology-based approach of context-sensitive computing for the optimization of flexible manufacturing systems (FMS) [20]. The approach addressed how to extract manufacturing contexts at source, how to process contextual entities by developing an ontology-based context model and how to utilize this approach for real-time decision making to optimize the key performance indicators (KPIs). Lim et al. utilized ontology for the modeling of product, asset, environment etc. in a digital twin-enhanced context-aware system for engineering product family design and optimization [21]. Chen et al. [22] and Wang et al. [23] also used ontology based techniques to build context models, with the former using Java Expert Shell System (JESS) and the latter, the Semantic Web Rule Language (SWRL) for reasoning.

In addition to aforementioned software-kind researches, context-aware hardware has also attracted attention. Lee et al. proposed an adaptive human machine interface (HMI) to provide only the most relevant information dynamically for users [24]. The WearIT@work project aimed at facilitating real-life industrial development of wearable technologies where a full technical realization scheme including wearable equipment, data exchange, and information display was proposed [25]. devices developed within such schemes provide more options to answer the *how* question, and can be considered as the basis for context-aware manufacturing system implementations.

The outcome of the research enumerated above shows that context-aware manufacturing systems have been identified as having potential benefits in many aspects. The development of the semantic web brings opportunity for explicit, sharable, and reusable context modeling and thus enhance the understandable ability of system through reasoning.

However, most of context-aware systems with reasoning ability are designed based on rules, even though ontology has its own basic rules for model consistency detection, additional rules are still needed for various context-aware problems. The existing rule-based context-aware manufacturing system usually focus on a specified application at the system design stage, some rules are embedded into the system programming, which is very difficult to update without shutdown. Though some rules are stored as a file (rules represented by ontology), manually update will result in a long update cycle and increase costs. Therefore, context-aware systems with pre-defined rules have challenges in adapting to the current volatile manufacturing environment.

The hypothesis in this paper is that for *when* decisions are required, the trigger can be extracted from historical context data without a long delay and cost. The resulting context-aware system would thus be more adaptive to volatilities in the manufacturing environment. It could be said that, the biggest difference between the proposed context-aware manufacturing system and the rule-based one is the rules' generation method and update mechanism.

2.2. Application of machine learning algorithms in manufacturing

Choudhary et al. pointed out that the accumulated manufacturing data makes manual analysis impractical, and thus data mining should be applied in manufacturing to get better quality information instead of vast amounts of raw data [26].

Machine learning is one of effective means for data mining. Pham et al. reviewed five types of machine learning techniques and their applications in manufacturing, such as quality control and scheduling problem [27]. Wuest et al. introduced three types of machine learning in this domain (supervised, unsupervised, reinforced), and focused on supervised machine learning algorithms identifying them as being particularly useful for monitoring, fault diagnosis, image recognition etc. [28].

Some researchers have conducted experiments with various algorithms in specific manufacturing fields and reported the comparative results: Lieber et al. compared the performance of three types of algorithms for quality prediction[29]. Paolo et al. reviewed six types of machine learning approaches applied in dynamic scheduling [30]. For the same application, Priore et al. compared three machine learning algorithms [31]. Loyer et al. compared five machine learning methods for estimating the manufacturing cost of civil jet engine components during the earliest phase of the design process [32]. Wu et al. applied random forest for cutting tool wear prediction [33], and also compared its performance with the performance of artificial neural network (ANN) and support vector regression (SVR). Similarly, Han et al. did research on tool wear compensation parameter recommendation for computer numerical control [34]. Linear regression and support vector machine (SVM) were used within the data mining software Weka to compare the results. Layer thickness prediction [35] and surface roughness prediction [36] are also manufacturing problems where a solution by SVR, regression analysis and neural network (NN) has been attempted. Aksoy et al. showed the application of inductive learning for acquisition of knowledge in manufacturing systems and demonstrated that they can achieve good performance [37].

In addition to supporting basic numeric or nominal data that most manufacturing systems generate, machine learning methods can also be applied for image processing through extracting the necessary set of rules and template matching techniques in industrial visual inspection systems [38].

The reviewed papers show that machine learning algorithms perform well in various manufacturing applications based on accumulated manufacturing history data. Context data of manufacturing, usually in a workshop can also be collected timely and stored in a database due to the development of sensors, computers, and Internet. Some relationships between the status of device and the product, or between the operation and the person are hidden in the accumulated historical data, which provide data support to utilize machine learning method for these implicit but useful information mining. If machine learning method can be effectively combined with historical context data to establish the rules or, indeed, a black-box model with hidden rules for a context aware system, then the aforementioned challenge of rule-based context awareness could be solved.

These papers also show that various machine learning algorithms have their strengths, weaknesses and appropriate application fields. Therefore, when applying machine learning to context-aware manufacturing system design, it is necessary to select the algorithms according to characteristics of specific context-aware problem and application.

3. A machine learning based context-aware manufacturing system

Since context model will guide database design and lead to some basic concept definition, it will be introduced firstly. Then, the framework of context-aware manufacturing system follows to show the composition of the system. At last, design of the most important module: context extraction module, which distinguishes a machine leaning based context-aware system and a rule-based one, will conclude the section.

3.1. Context model design

Dey's [1] well-accepted definition of context is general and should be refined in detail for the specific context-aware problem. An understandable context model can, not only, help users choose and install the right sensors, but also, assist in tuning of the context data storage to improve data access efficiency. Computer readable ontologies have been adopted by many researchers for context modelling because of their semantic richness and self-consistency. This paper focuses on the manufacturing context in a workshop and uses the ontology web language (OWL) to build a manufacturing context model with the core entities and corresponding properties as shown in Fig. 1.

Entity is something that exists separately from other things and has its own identity. In a workshop entity, entity could be a device, a person or a product and they could be classified more detailed such as device has subclasses of CNC machine tools, robot, automated guided vehicle (AGV) etc. It can be seen that context data comes from two entities in manufacturing, either from installed sensors on above mentioned entities or from the control system of device, which is consistent with actual physical connection. For context data, it has two basic data properties, meaning it has a certain value at a certain time. In contrast with other similar context models, entity *sensor* is added into the model to decouple the physical entity and the data. As a result, data can be retrieved through different dimensions such as a timestamp, a sensor, or even a device or person, which makes updates of the context extraction logic in the context-aware manufacturing system become clear. Beside properties shown in Fig. 1, *ID* and *location* are the basic properties that device, people and product should have. The *location* could be linked to them through sensors like GPS or radio frequency identification devices (RFID).

Fig. 1 only shows the concept design of context model, and every entity in the model needs instantiation by assigning the value of each basic data type. For example, temperature and humidity of workshop are two kinds of important contextual information, after initiating *Workshop*, two *Sensor* instances are instantiated and linked to a *Workshop* instance. Then, two *Data* entities are instantiated and linked to *Sensor* instances separately. The value collected by the physical sensor is assigned to *Data.value* and collection moment is recorded in *Data.dataTimeStamp*. This instantiated real-time workshop context is set as input for subsequent context extraction.

Though the design of context model is a two-dimensional model, when it is instantiated and stored in the database as part of historical context, it can be visualized as a cube (shown in Fig. 2), whose third dimension is time. That means context of each moment is a slice of the context cube, when there is a need to access the context at a specific time or the dynamic context over a certain time range, it is easy to retrieve the data from the database according to the timestamp.

3.2. System framework design

The Framework of machine learning based context-aware manufacturing system is shown in Fig. 3. Considering the whole flow of context data in and out the context-aware system, four main modules are designed with functions listed as follows:

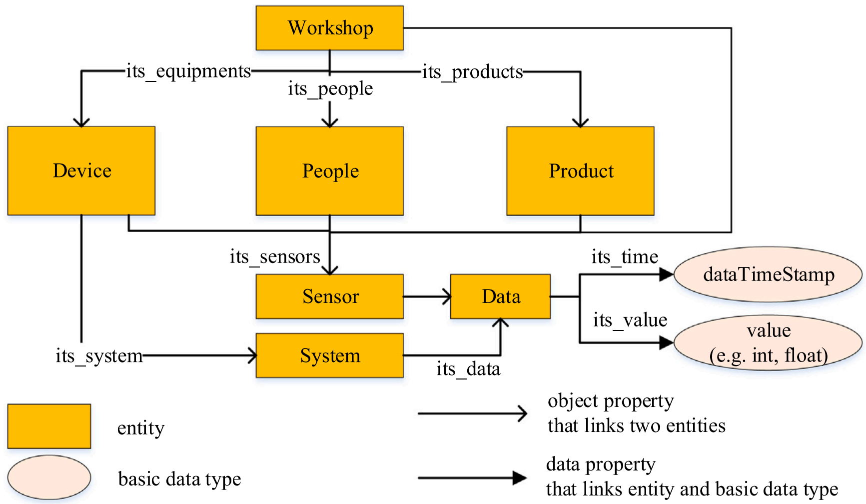

Fig. 1. Context model for manufacturing systems.

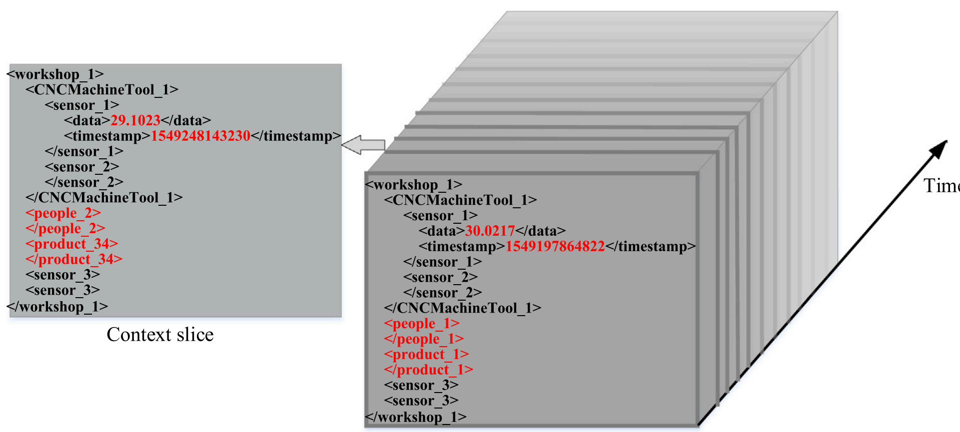

Fig. 2. Context cube.

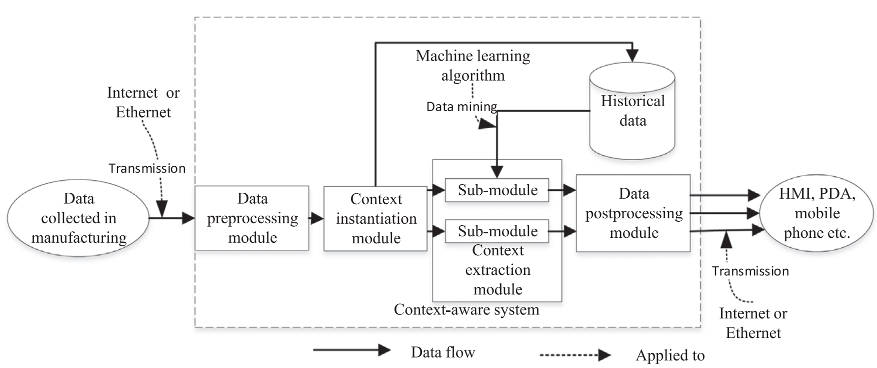

Fig. 3. Framework of context-aware manufacturing system.

- **Data preprocessing module** applies basic preprocessing techniques such as data cleansing and data transformation to the collected raw data.
- **Context instantiation module** uses real-time context data to instantiate context entities according to the context model and store them in a database as part of the training set for machine learning. Different parts of the context instances are also sent to corresponding sub-modules for further context extraction.
- **Context extraction module** consists of different sub-modules. Each sub-module is developed according to different application purposes and updates itself regularly based on the historical context data and suitable machine learning algorithms.
- **Data postprocessing module** combines the result of several sub-modules or directly dispatches them to the proper hardware depending on the context-aware purpose.

Each module is relatively independent and thus can be developed in parallel only with the exception of context extraction module. This is because data accumulation is the basis for machine learning.

The differences between the framework of proposed context-aware system and that of the rule-based one mainly are the database and context extraction module. In a rule-based context-aware manufacturing system, database is usually used for the storage of rules or expert knowledge, there would be a module responsible for accessing the database, retrieving rules and accepting the real-time context data to reason the useful information. In this designed framework, database stores historical data, which requires larger storage space. Rules, hidden in each sub-module, is generated and updated dynamically based on the historical data and machine learning algorithms. The process of model training and model usage do not affect each other, which allows sub-module updates during system running.

The framework starts from analyzing the manufacturing environment and instantiate the context model with consideration of each entity and their achievable status. Because the ontology-based context model can be updated at any time by removing or adding entities according to the changeable manufacturing environment, NoSQL database [39] is recommended rather than traditional relational database due to its character of modifying column dynamically without stop running. Even though the storage space of NoSQL database can be expanded using a larger computer cluster and duplicated data can be handled during data preprocessing, data sampling frequency of slow-changing data still need to be considered during data collection to reduce the computing and storage burden.

Data preprocessing module adopts any algorithms [40] that deal with data problems such as missing, abnormal and especially duplicated, though these algorithms are not necessary for all collected data. Beside above-mentioned data cleaning process, collected data sometimes need a transformation like unifying the units or nominalizing. The pre-processing algorithms that applied on a context data could be none or more than one, it depends on the data subsequent application. For example, the data acquired from CNC system like spindle speed or axis location can be utilized directly without preprocessing.

After data are collected and preprocessed, they should be mapped into context instance through the context imitation module. In this module, context instances are not only one, but several pieces that instantiate whole or part of the context model. Then, these context instances go to two destinations. One is the database, which stores all the historical data in a format of context cube and supports the training of sub-module of context extraction module. The other is the context extraction module, whose sub-modules accept different context instance, also known as context slice, as the input.

The context extraction module is a key part of machine learning based context-aware system. In this module, useful information is distilled from mass amount of real-time context data based on the trained sub-modules. Each sub-module may adopt different machine learning algorithm, take different context data as input and surely output different information. Therefore, selecting a suitable machine learning algorithm according to specified problem plays an important role in context extraction module design and significantly affects the performance of the sub-modules. The design of this module will be introduced detailed in the next section.

Finally, useful information generated by context extraction module go through the data postprocessing module. Multi-source information could be fused and calculated in this module according to the context-aware targets. A simple configuration file is established, which defines the information source and the calculation formula based on the result of each sub-module.

3.3. Context extraction module design

Nominal data and numerical data are the basic types of context data that can provide enough information for context-aware system in manufacturing. Each sub-module either takes nominal data as output, which means it solves a classification problem; or numerical data as output, which means it solves a regression problem.

The context extraction module is designed for not only answering the *when* question, but also other questions mentioned in Section 1. The illustrative example of work clearance in a factory can be used to give a better idea about the functionality of this module. Assuming that the location of each machine is mapped on a digital shop floor layout and that RFID are installed both on machines and, through wearables, on the workers. Through analyzing historical data, we can summarize that the worker who is nearest to each machine in the work sequence and then the machine turns off after he leaves must be responsible for work clearance. In such a situation, even if the responsible worker changes or the machine changes or moves, the context-aware system can quickly rebuild the digital map and push the new shut-down schedule to the right worker without additional rule modification. At least, two sub-modules need to be designed for role recognition and schedule generation separately. That means, the output of sub-module can only be one type and complex context-aware problem needs cooperation of several sub-modules.

The basic factors that need to be considered for module design are the input, the output and the inner logic. After deciding the output type, the input and the inner logic can both be acquired through the machine learning algorithm selection workflow as shown in Fig. 4.

For various output types, feasible algorithms are introduced as follows:

- Classification algorithms

  Kotsiantis enumerated 5 types of classification techniques [41]: logic-based algorithms, perceptron-based techniques or artificial neural networks (ANN), statistical learning algorithms, instance-based learning, and support vector machines (SVM).

  When applying these algorithms for context awareness in manufacturing, 3 points need to be noted. Logic-based algorithms have the best comprehensibility, while ANN cannot provide any classification logic for people to make decision making like adjusting parameters. Instance-based learning takes much more time for classification, which is not suitable for real-time job, especially when the data amount is very large. SVM's complexity is affected by data amount rather than feature amount; this makes it potentially suitable for context-aware problem based on short-term historical data with multiple features.
- Regression algorithms

  Regression analysis (RA) [29,31], neural networks [31] and SVR [29,30] are the commonly used algorithms for the regression problem. These algorithms can be applied for context-aware in manufacturing. It is noteworthy that the NN model is a black box, based on which, it is hard to provide information for further improvement.

There are many aspects for the performance comparison of machine learning algorithm, and in practical scenarios, a weight should be assigned to each comparison dimension according to the application. For a context-aware manufacturing system, the decision time is more important than the training time in most applications, because information of failure or safety problem need to be provided as quickly as possible.

Selection of attributes for the training set is the same as selection of input for the sub-module. Only the related part of the context instance is set as the input of sub-module and only related historical context data is selected as the training set, which helps to reduce computation burden and shorten the training and decision times. A relatively large range of attributes can be selected according to experts' experience and then it can be refined using feature selection algorithms [42].

Training set preprocessing also has a relationship with the application purpose. This means whether preprocessing is carried out or what preprocessing methods should be utilized depends on the comparison

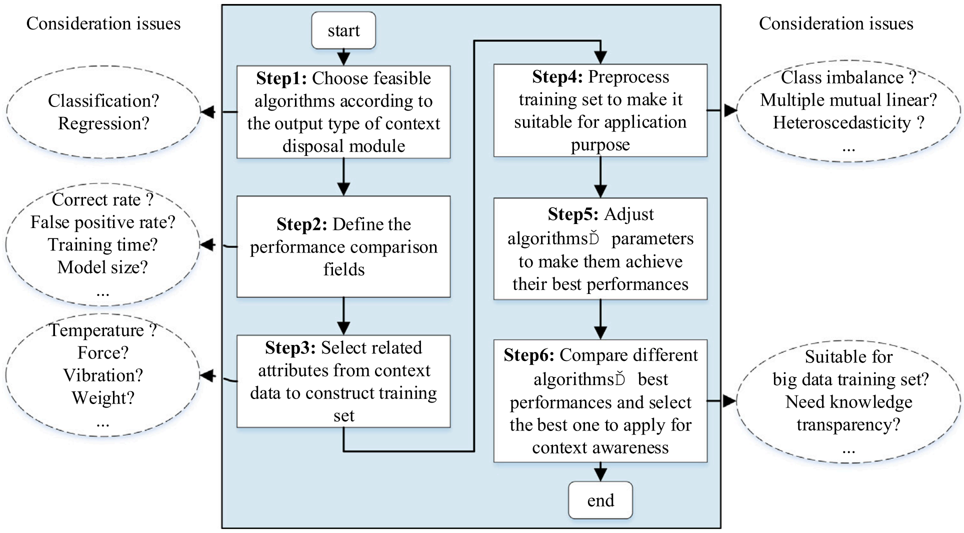

Fig. 4. Machine learning algorithm selection workflow.

fields that selected from the step 2.

When comparing the overall performances of machine learning algorithms, other factors that are not get from statistical analysis should also be taken into consideration. For example, a monitoring and alerting application does not necessitate knowledge transparency, whereas transparency is a basic demand for decision-making applications.

Finally, sub-module update mechanism also should be defined to make the context-aware system adaptive to a manufacturing environment that changes in an agile manner. The selection of either a suitable event trigger (product changes) or a suitable fixed time period (e.g. daily or weekly) can be made based on the parameters of the problem. And historical data access range also should be considered because it decides the size of training set, which should be within the range of computer processing ability.

4. Case study

Since the context extraction module is the key part that distinguishes a machine learning based context-aware manufacturing system from a rule-based one, this section will focus on verification of the context extraction sub-module through an example case.

4.1. Specified context-aware problem description

In this example, product quality prediction has been chosen as the application, a prototype implementation of the framework is then used to answer the *when* question using machine learning techniques. The selected problem is a subset of product quality control, as shown in Fig. 5, in the colored box. Here the historical context data are used to generate two different sets of training data according to the application purpose, which also known as the output of context extraction submodule. One training set is passed on to the sub-module that judges the production quality. While the other training set goes through a differently configured but similar unit to identify the person that responsible for quality issues.

When this context-aware manufacturing system runs, these two submodules should update themselves in a certain cycle according to historical context data and mining the required information according to real-time context data. But for the second sub-module, its running frequency is much lower than the first one, because people' role changes slowly.

4.2. Training set generation

In order to assess the viability of the proposed approach, a discrete event simulation model of a workshop is established using SIMIO software as shown in Fig. 6. This provides a platform for testing various scenarios at a low cost. In this simulation, Operator1 is responsible for machines CNC 1 and CNC 2, while operator 2 is responsible for machines CNC 3 and CNC 4. We took product quality control of CNC 2 as an example to illustrate the aforementioned machine-learning algorithm selection workflow. The context data related to CNC 2 are used to build the training set to focus on machine-level context awareness; corresponding states and properties are listed in Table 1. While it is possible to consider system-level context awareness, validation of the output would be much more difficult in comparison with a single machine and thus the simpler case has been chosen. Fig. 7.

Using model trace function in SIMIO, states with timestamps during the model running are exported in the format of CSV file which can then be taken into the machine learning environment. In order to have access

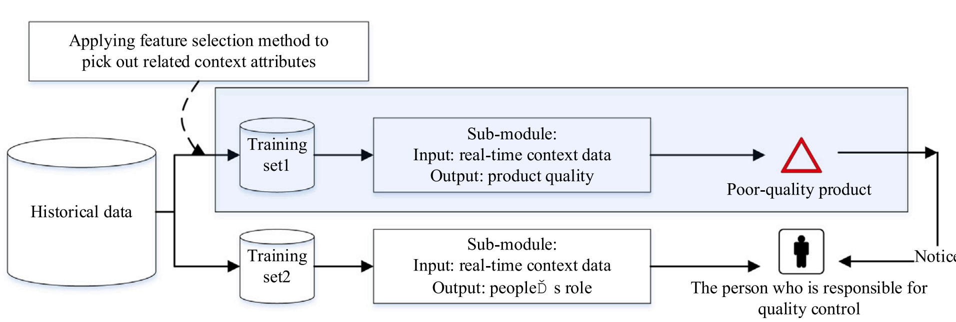

Fig. 5. Product quality control flow using machine learning method.

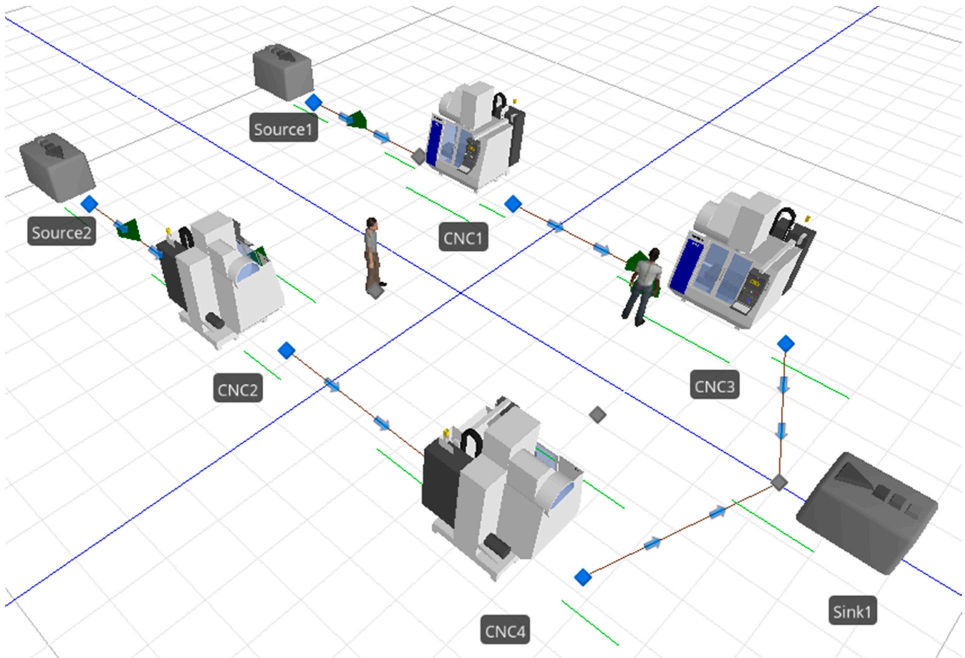

Fig. 6. An example manufacturing system modelled in Simio.

Table 1
Parameter setting.

Name	Type	Range	value	Definition
TH_temperature	Property	CNC2	40	Threshold of warning for CNC2's spindle temperature
TH_vibration	Property	CNC2	20	Threshold of warning for CNC2's cutting tool vibration
Temperature	State	CNC2	Random. Triangular (CNC2. Threshold −7, CNC2. Threshold −5, CNC2. Threshold +0.5)	Simulated CNC2's spindle temperature
Vibration	State	CNC2		Simulated CNC2's cutting tool vibration
Quality	State	CNC2	True (1, good) or False (0, bad)	Simulated CNC2's product quality
TH_humidity	Property	Model	19	Threshold of warning for shop floor's humidity
TH_temperature	Property	Model	32	Threshold of warning for shop floor's temperature
Humidity	State	Model	Random. Triangular (Model. Threshold −7, Model. Threshold −5, Model. Threshold +0.5)	Simulated CNC2's shop floor's humidity
Temperature	State	Model		Simulated CNC2's shop floor's temperature

For the behaviors and properties configuration of CNC 2 in SIMIO, random values to the four states except quality are assigned in the process of *CNC2_Processing*. Then in the process of *CNC2_AfterProcessing*, quality state is determined according to the decision flow in Fig. 7.

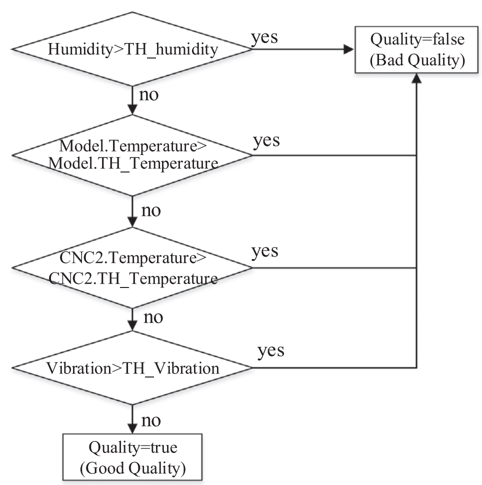

Fig. 7. Quality state decision flow.

to various machine learning algorithms in an easy deploying environment, Weka [43] has been used. Weka provides a Java API to convert CSV formatted data to its native ARFF format, Fig. 8 shows an excerpt of the generated file, called the *basic training file*. Each training file represents one training set. All context data in each line are listed according to timestamp, but timestamps are removed from the *basic training file* because it is not relevant for the product quality prediction in this case.

To eliminate the casualty, totally 10 *basic training files* are generated using different random seed to grantee data unique. After statistic calculation, the instances amount in each file is around 4600 and the ratio between good quality and bad quality is around 45:1.

Data in *basic training files* represents what is sensed in an ideal manufacturing environment. But in practice where context data are collected by sensors, the veracity of the information can be affected by the harsh environment leading to missing data or noise. This is simulated by replacing the values in the *basic training files* with missing items, the missing percentage is set from 0.01 to 0.05. Noise is also added to the *basic training files* to replace normal values with abnormal ones through using a white noise generation function with a signal to noise ratio (SNR) of 10.

```
@relation basicfile_seed1

@attribute SF_huminity numeric
@attribute SF_temperature numeric
@attribute Temperature numeric
@attribute Vibration numeric
@attribute Quality {Good,Bad}

@data
14.554538,28.989418,36.02353,16.447739,Good
15.858462,28.890602,38.00164,15.585195,Good
15.858635,31.445587,36.868415,13.336456,Good
17.992763,27.01735,34.577039,17.363952,Good
16.094073,30.406978,35.127236,17.782379,Good
13.650973,28.242047,37.389068,15.575354,Good
15.141388,26.104185,34.717708,15.579513,Good
17.963834,27.681348,35.842076,15.246493,Good
17.258237,27.806824,35.82888,14.749811,Good
```

Fig. 8. Fragment of ARFF file.

4.3. Machine learning algorithm selection

Step 1 Choosing feasible machine learning algorithms.

Through analyzing the literature reviewed in Section 2.2, five frequently-used and representative types of machine learning algorithms for the classification problem are selected as listed in Table 2. Though they may have improved algorithms with better performances, this case just shows the necessary of machine learning algorithm selection, and find the best one from relative basic options.

Step 2 - Define the performance comparison criteria.

The purpose of poor-quality detection task is to reduce the misclassified poor-quality product (having a low False Positive rate) rather than increasing the correct classification rate. Because manufacturers care about implications of providing customers a poor-quality product without detecting the flaw as opposed to detecting a good-quality product as bad. So, the False Positive (FP) rate (in this case, poor-quality stands for negative side, while good-quality stands for positive side) is chosen as the performance comparison criterion for the tested machine learning algorithms.

Step 3 - Select the attributes related to performance criteria to build the training set.

In this experiment, four attributes that relate to quality classification are known. However, when using data from actual manufacturing procedure, related attributes need to be picked for a specific classification problem from various context data. If used Weka, *InfoGainAttributeEval* algorithm could be used for the selection of high-related features because it can calculate correlation between feature and class. Consequently, only high correlation features are chosen to participate in the classification, which not only increases the classification performance but also improves the computational efficiency.

Step 4 - Preprocess the training set.

In a generated training set, similar to a well-calibrated actual manufacturing system, the ratio between good-quality products and poor-quality products is very large and this leads to the class imbalance problem. This problem indicates that a classification algorithm is trying to divide a set into a number of subsets (two here) where one is much larger than the other one. This imbalance can lead to difficulties in convergence. Whether solving this problem depends on the degree of class imbalance and the selected performance comparison criteria. Thus, a pre-experiment is conducted to check whether such degree of class imbalance affects FP rate, and also for other application purposes that utilize correct rate as performance comparison criteria.

SMOTE (Sampling a synthetic minority over sampling technique) algorithm [44] is applied to balance the class amount through increasing the minority amount. To avoid over-fit, 10-folds cross-validation is applied in this experiment. Table 3 listed the average performance of selected machine learning algorithms. Correct rate reduces a little after using SMOTE, which we can accept, while FP rate decreases dramatically as was hoped. So, SMOTE will be used for the remaining experiments in this paper.

Step 5 - Adjust parameters of each machine learning algorithm to achieve its best performance.

Different parameter settings of machine learning algorithms would affect its performance a lot[45]. However, due to time limits, it is difficult to tune parameters of each machine learning algorithm for its best performance. As shown in Table 2, IBK and LIBSVM use different parameters in this case to check their influences on the algorithms' performance and show the necessity of parameter adjustment.

4.4. Result analysis

4.4.1. Algorithm performance for the basic training set

Fig. 9 shows the performance of the difference algorithms for 10 *basic training files*. ANOVA (Analysis of Variance) is applied to verify that there was a statistically significant performance difference between the algorithms. The result is shown in Table 4. F value is high enough to show that the different algorithms have different classification performance for this context-aware problem and thus there is a need for algorithm selection.

From the comparison of IBK (K is different) and LIBSVM (kernel is different) with different parameters, it can be seen that a larger K helps IBK perform better and a polynomial kernel helps LIBSVM perform better. This also verifies that parameter tuning really affects the machine learning algorithms' performances and thus the step 5 is necessary.

Over the whole, J48, Naïve Bayes and LIBSVM with POLY kernel perform much better than other algorithms.

Beside FP rate comparison, the training time and testing time are also analyzed. Because training times vary widely, which cannot be illustrated clearly in a chart, Table 5 is used to show the average training time of the 10 training sets using different machine learning algorithms.

It can be seen that LIBSVM with POLY kernel takes much more time

Table 2
Algorithms setting.

Name in Weka	Main Para Set	Sub-Type	Type
J48		Decision Tree (C4.5)	Logic-based learning
IBK	K = 1 K = 5	K-Nearest Neighbour	Instance-based learning
Naïve Bayes	UseKernalEstimator=True	Naïve Bayes	Statistical learning
LibSVM	Kernel = polynomial (POLY) Kernel = radial basis function (RBF)		SVM
Multilayer perception	Hidden Layer = 5		Perceptron-based techniques

Table 3
Product quality prediction result between with SMOTE and without SMOTE.

Machine learning algorithm	Correct rate	Correct rate using SMOTE	FP rate	FP rate using SMOTE
Naïve Bayes	99.932	97.064	0.016	0.005
IBK = 1	97.646	95.934	0.727	0.29
IBK = 5	97.77	94.495	0.951	0.198
LIBSVM(POLY)	97.845	94.352	0.686	0.02
LIBSVM(RBF)	97.752	97.035	0.982	0.148
Multilayer perception = 5	98.219	91.587	0.741	0.154

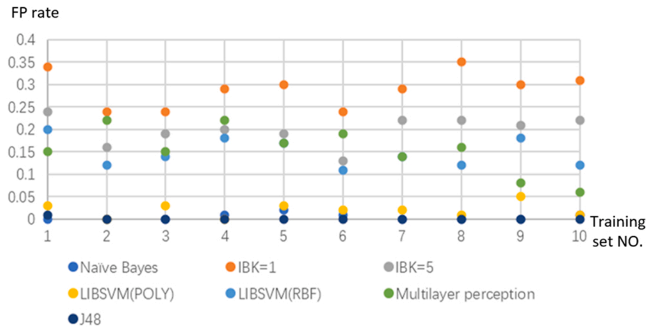

Fig. 9. FP rate result for the basic training set.

Table 4
ANOVA result to show algorithm performance difference.

Source	SS	df	MS	F	Prob>F
Columns	0.74228	6	0.12371	128.82	1.99788e-33
Error	0.0605	63	0.00096		
Total	0.80278	69			

than other algorithms. Even taking this longest training time as the update time for one rule, using machine learning to update rules is still faster than manually modifying rules.

Testing time is the more important comparison criteria than training time in the context-aware manufacturing system. It means after sub-modules being trained well, how long it will take to give the result based on the real-time context data. It could be seen from Fig. 10 that IBK performs the worst among all the algorithms. Considering the control periods of CNC machine tools or robots usually are in milliseconds, the instance-based learning is not suitable for urgent context extraction in manufacturing.

4.4.2. Algorithm performance for the missing value training set

Fig. 11 shows the missing value effect on algorithms' performances. The horizontal axis is missing value rate, while corresponding value is the average of 10 training sets. With missing value amount increasing, all algorithms performances decrease. J48, Naïve Bayes and LIBSVM with POLY kernel are still better than other algorithms.

4.4.3. Algorithm performance for the noisy training set

Fig. 12 shows the performance of algorithms for the 10 training sets with noise. This noise is not same to the noise in machine learning (usually the wrong classification) but is quite common in manufacturing (e.g. a wrong monitored value resulted from disturbed electrical signal). Compared with Fig. 9, all algorithms' performances are affected by the noise, especially J48 and Naïve Bayes.

4.4.4. Algorithm performance for large-scale data and knowledge transparency

Theoretically, the more data in the training set, the higher the correct rate an algorithm achieves. Though distributed computing technology in cloud manufacturing can solve large-scale data problem, some algorithms need to load the whole data set in the memory, which is restricted by the memory of a single computer, e.g. J48. However, J48 provides the clearest classification standard as shown in Fig. 13, which can help users make more definite decisions. Take product quality control for example, after the sub-module of product quality prediction is built, thresholds of each state that lead to poor-quality are known. Then, when the sub-module judges a product is poor-quality according to real-time context data, only abnormal states will be provided to the related

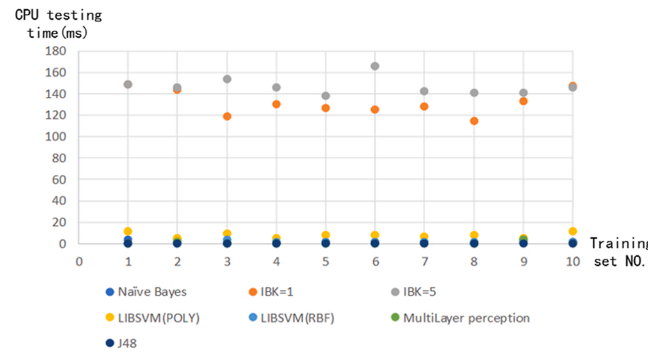

Fig. 10. CPU testing time for the basic training set.

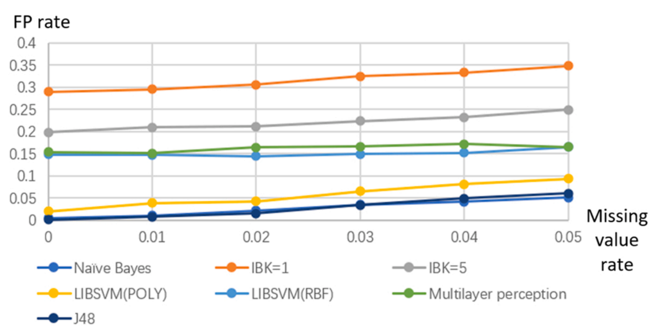

Fig. 11. FP rate result for the training set with missing values.

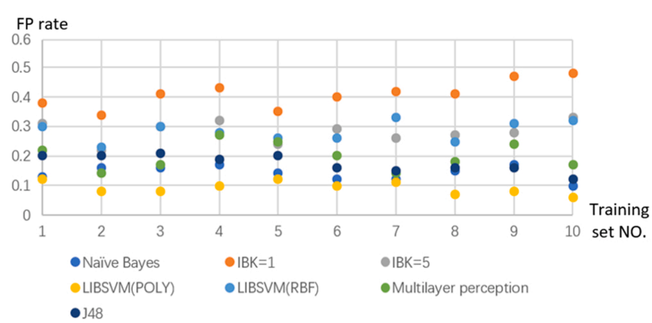

Fig. 12. FP rate result for the training set with noise.

```
Temperature <= 39.999658
|   Vibration <= 19.997511
|   |   SF_huminity <= 18.999351
|   |   |   SF_temperature <= 31.951899: Good (4501.0)
|   |   |   SF_temperature > 31.951899
|   |   |   |   SF_temperature <= 31.98265: Good (7.0)
|   |   |   |   SF_temperature > 31.98265: Bad (18.0)
|   |   SF_huminity > 18.999351: Bad (25.0)
|   Vibration > 19.997511: Bad (28.0)
Temperature > 39.999658: Bad (31.0)
```

Fig. 13. Classification logic of J48.

Table 5
Average training time of each machine learning algorithm (unit: ms).

Naïve Bayes	IBK = 1	IBK = 5	LIBSVM (POLY)	LIBSVM (RBF)	MultiLayer perception	J48
3.126	0.938	0.937	12,580	93.336	2100.075	13.907

people and these states can be adjusted according to the classification standard.

4.4.5. Overall performance

For the product quality control problem in this case, J48 performs almost the best one because the simulated context data is generated by rules, which is suitable for a logic-based machine learning algorithm to exploit its performance. Besides, its outstanding advantage is the good knowledge transparency. It should be noted that, the actual context-aware question in manufacturing maybe not a logical one as simulated. And if the manufacturing environment is very harsh, the drawback of J48 that it is sensitive to noise would make its performance inferior to that of LIBSVM with POLY kernel.

Then, the trained classifier with a suitable algorithm can be embedded as sub-module in the context-aware manufacturing system, with four context data as inputs and product quality as output. If J48 was utilized, abnormal context data and its corresponding normal range can also be provided for users to help them make decision efficiently. For this case, the hybrid update mechanism that utilizes both certain cycle and events is suitable, e.g. when product changes, sub-module must be updated. Otherwise, update frequency of once a month is enough because CNC machine tools performance won't have obvious degradation in a short time.

We also analyze different algorithms' performances on the correct rate and compare the result with that in [30]. Though the correct rate does not affect performance analysis in this case, it shows two general rules as follows.

- Algorithms' performance on FP rate has no relation with their performances on correct rate, neither in direct proportion, nor in inverse proportion (e.g. LIBSVM with POLY kernel performs better than LIBSVM with RBF kernel in the FP rate, while for the correct rate, LIBSVM with RBF kernel performs better. J48 performs best both in the correct rate and the FP rate).
- Algorithms' performances are affected by training set (e.g. decision tree is not the best one in research [30]), only if training sets have common characteristics, the performance comparison results have reference value.

In this case, trigger context that leads to poor-quality products can be extracted from historical context data in a few minutes and detailed decision-making information can be provided for users by adopting the J48 algorithm. When product changes, this machine-learning based context-aware manufacturing system can update context extraction module efficiently and automatically by applying the selected algorithm to new context data. Then product quality control strategies can be adjusted in a timely manner to guarantee stable production. While rule-based system needs more time to define new rules manually and the coupling between programming and rules during system development make this update more difficult.

5. Conclusions

This paper concludes the four questions that need to be answered in a context-aware system. Aiming at the manufacturing filed, machine learning algorithms are utilized in context-aware manufacturing system design to make it adaptive for current changeable market and thus can answer the four questions with less human intervention.

This paper refined context in the manufacturing area, which considers human factor and the decoupling between data and entity. This context model could guild the manufacturing context data collection and access. Then, based on this model, the framework of context-aware manufacturing system is designed, which consists of data preprocessing module, context instantiation module, context extraction module and data postprocessing module. Since context extraction module plays the most important role to distinguish the machine learning-based context-aware manufacturing system and rule-based one, a common machine learning algorithm selection workflow is introduced for this module design with consideration of characteristics of manufacturing context and actual application targets. At last, the *when* question of product quality control is answered to verify the feasibility of applying machine learning algorithms in a context-aware manufacturing system and show the necessity of machine learning algorithm selection.

Further research will focus on answering other context-aware questions, especially the *who* question, which needs training set including timestamps to find motion logic under context data. Deep leaning and incremental learning will be researched for such complex context-aware application.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The work is supported by the National Natural Science Foundation of China (Grant No. 51875323) and the Key Research and Development Project of Shandong Province, China (Grant No. 2019JZZY020121).

References

[1] Brunoe TD, Soerensen DG, Nielsen K. Modular design method for reconfigurable manufacturing systems. Procedia CIRP 2021;104:1275–9.
[2] Latsou C, Erkoyuncu JA, Farsi M. A multi-objective approach for resilience-based system design optimisation of complex manufacturing systems. Procedia CIRP 2021;100:536–41.
[3] Herps K, et al. A simulation-based approach to design an automated high-mix low-volume manufacturing system. J Manuf Syst 2022;64:1–18.
[4] Leng J, et al. Digital twins-based remote semi-physical commissioning of flow-type smart manufacturing systems. J Clean Prod 2021;306:127278.
[5] Dey AK, Abowd GD. Providing architectural support for building context-aware applications. Ga Inst Technol 2000.
[6] Dey AK. Understanding and using context. Pers Ubiquitous Comput 2001;5(1):4–7.
[7] Stöttinger M, Context-Awareness in industrial environments; 2004.
[8] Atzori L, Iera A, Morabito G. The internet of things: a survey. Comput Netw 2010;54(15):2787–805.
[9] Singh S, Vajirkar P, Lee Y. Context-based data mining using ontologies. In: Conceptual modeling - ER 2003. Berlin, Heidelberg: Springer Berlin Heidelberg; 2003.
[10] Danylenko A, Kessler C, Löwe W. Comparing machine learning approaches for context-aware composition. In: Software composition. Berlin Heidelberg: Springer Berlin Heidelberg; 2011.
[11] Matei O, et al. Context-aware data mining: embedding external data sources in a machine learning process. in hybrid artificial intelligent systems. Cham: Springer International Publishing; 2017.
[12] Ren Y, Li G. A contextual sensor system for non-intrusive machine status and energy monitoring. J Manuf Syst 2022;62:87–101.
[13] Safdar Ali, Anna Lewandowski, Joerg Rett. A SOA based context-aware order picking system for warehouses using Laser Range Finder and wearable computer. In: Proceedings of the 2011 IEEE international symposium on a world of wireless, mobile and multimedia networks; 2011.
[14] Ciccarelli M, et al. Human work sustainability tool. J Manuf Syst 2022;62:76–86.
[15] Shadbolt N, Berners-Lee T, Hall W. The semantic web revisited. IEEE Intell Syst 2006;21(3):96–101.
[16] Yl A, Mra B. Semantic communications between distributed cyber-physical systems towards collaborative automation for smart manufacturing - ScienceDirect. J Manuf Syst 2020;55:348–59.
[17] Hynes G, et al. SAGE: an ambient intelligent framework for manufacturing. IFAC Proc 2006;39(4):339–43.
[18] Zhiyu Chen, Shufen Liu, Xiao Wang, Application of context-aware computing in manufacturing execution system. In: Proceedings of the 2008 IEEE international conference on automation and logistics; 2008.
[19] Eirinakis P, et al. Situation-aware manufacturing systems for capturing and handling disruptions. J Manuf Syst 2021;58:365–83.
[20] Kamal Uddin M, et al. Ontology-based context-sensitive computing for FMS optimization. Assem Autom 2012;32(2):163–74.
[21] Kyhl A, et al. A digital twin-enhanced system for engineering product family design and optimization. J Manuf Syst 2020;57:82–93.
[22] Chen H. An intelligent broker architecture for context-aware systems. Phd Propos Comput Sci 2003.

[23] Wang XH et al. Ontology based context modeling and reasoning using OWL. In: Proceedings of the second IEEE annual conference on pervasive computing and communications workshops; 2004.
[24] Nieto Lee A, Martinez Lastra JL. Enhancement of industrial monitoring systems by utilizing context awareness. in 2013 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA). 2013.
[25] Lukowicz P, et al. WearIT@work: toward real-world industrial wearable computing. IEEE Pervasive Comput 2007;6(4):8–13.
[26] Choudhary AK, Harding JA, Tiwari MK. Data mining in manufacturing: a review based on the kind of knowledge. J Intell Manuf 2008;20(5):501.
[27] Pham DT, Afify AA. Machine-learning techniques and their applications in manufacturing. Proc Inst Mech Eng, Part B: J Eng Manuf 2005;219(5):395–412.
[28] Wuest T, et al. Machine learning in manufacturing: advantages, challenges, and applications. Prod Manuf Res 2016;4(1):23–45.
[29] Lieber D, et al. Quality prediction in interlinked manufacturing processes based on supervised & unsupervised machine learning. Procedia CIRP 2013;7:193–8.
[30] Priore P, et al. Dynamic scheduling of manufacturing systems using machine learning: an updated review. Artif Intell Eng Design Anal Manuf 2014;28:83–97.
[31] Priore P, et al. A comparison of machine-learning algorithms for dynamic scheduling of flexible manufacturing systems. Eng Appl Artif Intell 2006;19(3):247–55.
[32] Loyer J, et al. Comparison of machine learning methods applied to the estimation of manufacturing cost of jet engine components. Int J Prod Econ 2016;178:109–19.
[33] Wu D, et al. A comparative study on machine learning algorithms for smart manufacturing: tool wear prediction using random forests. J Manuf Sci Eng 2017;139:7.
[34] Han J, Kim R, Chi S. Applications of machine learning algorithms to predictive manufacturing: trends and application of tool wear compensation parameter recommendation. In: Proceedings of the 2015 international conference on big data applications and services; 2015. Jeju Island, Republic of Korea: Association for Computing Machinery.
[35] Benjamin Lenz, Bernd Barak. Data mining and support vector regression machine learning in semiconductor manufacturing to improve virtual metrology. In: Proceedings of the 2013 46th Hawaii international conference on system sciences; 2013.
[36] Feng CJ, Yu ZS, Kusiak A. Selection and validation of predictive regression and neural network models based on designed experiments. IIE Trans 2006;38(1):13–23.
[37] Witkowski T, Antczak P, Antczak A. Machine learning — Based classification in manufacturing system. In: Proceedings of the 6th IEEE international conference on intelligent data acquisition and advanced computing systems; 2011.
[38] Aksoy MS, Torkul O, Cedimoglu IH. An industrial visual inspection system that uses inductive learning. J Intell Manuf 2004;15(4):569–74.
[39] Moniruzzaman ABM, Hossain SA. NoSQL Database: new era of databases for big data analytics - classification, characteristics and comparison. Computer Science; 2013.
[40] Kong T, et al. Data construction method for the applications of workshop digital twin system. J Manuf Syst 2021;58:323–8.
[41] Kotsiantis SB. Supervised machine learning: a review of classification techniques. In: Proceedings of the 2007 conference on emerging artificial intelligence applications in computer engineering: real word AI systems with applications in eHealth, HCI, Information Retrieval and Pervasive Technologies; 2007: IOS Press.
[42] Guyon I, Elisseeff A. An introduction to variable and feature selection. J Mach Learn Res 2003;3(null):1157–82.
[43] Holmes Geoffrey, Donkin Andrew, Witten Ian. WEKA. A machine learning workbench. In: Proceedings of ANZIIS '94 - Australian New Zealnd intelligent information systems conference; 1994.
[44] Chawla NV, et al. SMOTE: synthetic minority over-sampling technique. J Artif Intell Res 2002;16(1):321–57.
[45] Weerts HJ, Mueller AC, Vanschoren J. Importance of tuning hyperparameters of machine learning algorithms. arXiv Prepr arXiv 2020. 2007.07588.