An Efficient Complex Event Processing Algorithm based on Multipattern Sharing for Massive Manufacturing Event Streams

Jianhua Wang[1,2], Yubin Lan[2], Shilei Lu[1*], Lianglun Cheng[3]
[1] College of Electronic Engineering, South China Agricultural University
Guangzhou, - P. R. China
[e-mail:wangjianhua655@163.com]
[2] National Center for International collaboration Research on precision
Agricultural Aviation Pesticides Spraying Technology,
South China Agricultural University, Guangzhou, - P. R China
[3] College of Automation, Guangdong University of Technology
Guangzhou, - P. R. China
[e-mail: 1280137559@qq.com, ylan@scau.edu.cn, lvshilei@scau.edu.cn, llcheng@gdut.edu.cn]
*Corresponding author: Shilei Lu

Received August 23, 2017; revised November 6, 2017; revised March 5, 2018; accepted July 2, 2018; published March 31 2019

Abstract

Quickly picking up some valuable information from massive manufacturing event stream usually faces with the problem of long detection time, high memory consumption and low detection efficiency due to its stream characteristics of large volume, high velocity, many variety and small value. Aiming to solve the problem above for the current complex event processing methods because of not sharing detection during the detecting process for massive manufacturing event streams, an efficient complex event processing method based on multipattern sharing is presented in this paper. The achievement of this paper lies that a multipattern sharing technology is successfully used to realize the quick detection of complex event for massive manufacturing event streams. Specially, in our scheme, we firstly use pattern sharing technology to merge all the same prefix, suffix, or subpattern that existed in single pattern complex event detection models into a multiple pattern complex event detection model, then we use the new detection model to realize the quick detection for complex events from massive manufacturing event streams, as a result, our scheme can effectively solve the problems above by reducing lots of redundant building, storing, searching and calculating operations with pattern sharing technology. At the end of this paper, we use some simulation experiments to prove that our proposed multiple pattern processing scheme outperforms some general processing methods in current as a whole.

Keywords: Complex event processing, Multipattern sharing, Massive event streams, Internet of manufacturing things

The work was supported by the National Natural Science Foundation of China(No.61602187) and and (No.61601189), the National Key Research and Development Plan (No.2016YFD0200700); the science and technology projects in Guangdong Province (No.2016A020209007) and (No.2016A020210088), the project of Natural Science Foundation of Guangdong Province (No. 2016A030310453) and the Guangzhou Science and Technology Project (N0.201707010482).

http://doi.org/10.3837/tiis.2019.03.015 **ISSN : 1976-7277**

1. Introduction

Internet of Manufacturing Things (IOMT)[1] is an important means to realize the informationization and intellectualization for the traditional manufacturing process, which can realize reliable perception, real-time transmission, pervasive calculation, precise control and trusted service in the whole manufacturing process of production design, manufacture, service, and other aspects by combining the electronic information technology and manufacturing technology, such as network, embedded system, RFID, sensor and actuator, and so on, and can improve its added value of production, enhance its control capability of manufacturing, and strengthen its service process [2].

However, in the actual manufacturing environment, since the increasingly large manufacturing scale, the more and more complex manufacturing process, the temporal and spatial distribution of production process, the multi-source interference of production environment, and so on, make many sensing equipments, such as RFID tags, sensor nodes, and so on, are deployed in large numbers to the manufacturing filed to monitor various manufacturing objects, for example, people, materials, equipments, production process, products, services and environmental changes[3]. These sensing devices collect various kinds of event in a quick and automatic way, and form massive manufacturing event streams.

Because these massive manufacturing event streams have following characteristics: 1) large volum, it can reach TB or even PB scale per second; 2) high velocity, it requires fast processing and response to the massive manufacturing event streams. 3) many variety, it includes many different event contents, such as people, material, equipment, process, product and service, and so on; 4) small value, there is repetitive and small amount of information between adjacent events, etc, that make it difficult to quickly find out some valuable information from them, thus affecting its extensive application. The existing processing methods about event streams cannot fully support the real-time processing for the massive manufacturing event streams.So how to quickly obtain some valuable information from massive manufacturing event streams become a very important challenge when processing the massive manufacturing event streams. Because Complex Event Processing (CEP)[4] technology can quickly pick up valuable information from a massive manufacturing event stream by using the association between event attributes, matching rules and algebraic operations, it has recently obtained a increasing attention in the field of event stream processing. In this paper, we mainly use complex event processing technology to quickly handle the massive manufacturing event streams above.

At present, many researches on complex event processing have been carried out to detect a complex event from event streams. In the processing system aspects, such as Cayuge system [5], Esper system [6], Estream system[7], PQS system[8],and so on, have been developed to detect a complex event from event streams; In the processing method aspects, four general complex event processing methods, such as complex event processing method based on diagram[9], complex event processing method based on tree[10], complex event processing method based on finite automaton[11], complex event processing method based on petri-net[12], and some of their improved methods, for example, complex event processing method based on timed petri-net[13], complex event processing method based on optimized directed graph[14], complex event processing method based on compressed composition tree[15], complex event processing method based on pushdown automata[16], and so on, have been developed to detect a complex event from event streams.

However, the existing processing systems or methods can only detect a single complex event from event streams at a time, but cannot realize the sharing detection for multiple complex events at the same time. Detecting multiple different complex events usually needs to construct multiple different detection models of complex event, which can lead to long detection time, high memory consumption and low event throughput due to not sharing detection. In real life, we also usually need to detect multiple different complex events with a general detection model from even streams at the same time. Under such a condition, if we still directly use the current processing methods above to detect multiple different complex events from massive manufacturing event streams, there will be long detection time, high memory consumption and low event throughput due to not sharing detection, and a large number of redundant building, storing, searching and calculating operations are used to construct for many the same prefix, suffix, or subpattern existed in the current complex event models, thus influencing its whole processing efficiency. In this paper, we call the current complex event processing methods as single pattern complex event detection methods.

In this paper, aiming to solve the problems above, an efficient complex event processing algorithm based on multipattern sharing is presented in this paper. The achievement of this paper lies that a multipattern sharing technology is successfully used to realize the high-efficient detection for multiple different complex events from massive manufacturing event streams with our scheme. Specially, in our scheme, we use pattern sharing technology to successfully merge many the same prefix, suffix, or subpattern, which exists in many single pattern complex event models and cannot be shared detection by them, into a multiple pattern detection model of complex event, and realize the quick detection for mutiple differnt complex events with the use of new detection model from massive manufacturing event streams, which can reduce lots of redundant building, storing, searching and calculating operations for them. As a result, the problems above existed in single pattern detection methods in current can be effectively solved by our scheme. The simulation results show that our proposed multipattern complex event processing scheme in this paper outperforms some general complex event processing methods in current as a whole.

The rest of this paper is organized as follows. In section 2, the related works of complex event processing are introduced. Our proposed complex event processing method is presented in section 3. The experimental results and analysis with our proposed scheme are shown in section 4. In section 5, we give some our conclusions.

2. Related Works

In recent, many related researches on complex event processing have been taken for the detection of complex events. In the processing systems aspect, Cornell University developed a Cayuge system[5], which mainly used customized automatic machine to realize the detection of a complex event from event stream. Esper Tech company developed an Esper system[6], which mainly used to the event match rules, event type inheritance, graph structure of object, event dynamical property, and so on, to realize the complex event detection from real-time event stream. The Arlington of Texas University developed an Estream system[7], which mainly took advantage of the integrated event interrogators and predefined rules to complete the detection of a complex event from event stream. Dartmout University developed a PQS system[8], which mainly used NFA(Nondeterministic Finite Automaton) and invisible markov technology to realize the detection of a complex event from event stream. The Berkeley of California University developed a SASE system[17], which mainly used AIS(Active Instance Stack) to realize the detection of a complex event for real-time RFID

event stream.

In the detection method aspect, four general complex event processing methods, for example, Viatkin et al.[9] presented a complex event detection method based on petri-net. Bai et al.[10] proposed a complex event detection method based on graph. Sun et al.[11] suggested a complex event detection method based on tree. Mei et al.[12] proposed a complex event detection method based on finite automaton, and some of their improved processing methods, for example, Jin et al.[13] proposed a complex event processing method based on timed petri-net. Wang et al.[14] presented a complex event processing method based on optimized directed graph. Li et al.[15] proposed a complex event processing method based on condensed composition tree. Cao et al.[16] proposed a complex event processing method based on pushdown automata structure, and so on, have been developed to detect a complex event from various event streams. In this paper[17], Eugene et al proposed a high-performance complex event processing method, called SASE, for real-time RFID stream. Zhang et al.[18] proposed a complex event processing method based on improved matching tree structure to detect a complex event from distributed uncertain event stream. Kyoung et al.[19] proposed a complex event processing scheme based on minimum conditions to detect a complex event from event stream. Wang et al[20] developed a complex event processing method based on multi DAG for multi-probability RFID event stream. Peng et al.[21] presented a scalable event processing method based on NFA state to detect a complex event from event stream. In this paper[22], Wang et al proposed a complex event processing method based on probabilistic NFA and AIS to detect a complex event for single distributed probabilistic event streams. In this paper[23], Peng et al. proposed a complex event processing scheme based on event selectivity to detect a complex event for live archived event streams. In the paper[24], Moon et al. developed an RFID business aware framework based on business rules to detect a complex event. In this work[25], a complex event processing algorithm based on RCEDA was proposed to detect a complex event. In the paper[26], a complex event processing engine, called DSMS was proposed to detect a complex event from the massive RFID stream. In the paper[27], an efficient complex event processing algorithm based on INFA-HTS (Improved Nondeterministic Finite Automaton-Hash Table Structure) was proposed to detect a complex event for out-of-order RFID event streams. In this paper[28], a new parallelization model and three parallel processing strategies were proposed for distributed complex event processing systems to address the difficulties of implementing parallel processing. In this paper[29], a distributed query-plan of complex event processing structure and algorithm based on directed acyclic graph was developed to solve the problem of a single complex event or a small quantity of events. In this paper[30], an intelligent complex event processing method with D numbers under fuzzy environment was proposed to address the issues of intrinsic uncertainty in pattern rules. In this paper[31], a novel complex event detection based on joint max margin and semantic was presented to address the limitations of semantic and temporal features. In this paper[32], a series of optimized algorithms based on nondeterministic automata model were used to reduce its bottlenecks and to realize the detection of complex event after analyzing its complexity of expensive queries of complex event.

However, the existing processing systems and methods above can only detect a single complex event from event streams once a time, but cannot realize the detection for multiple complex events from event streams at the same time due to not pattern sharing. Detecting multiple different complex events usually needs to construct multiple different complex event detection models, which can cause long detection time, high memory consumption and low event throughput due to not sharing the same prefix, suffix or subpattern in them, thus affecting their whole detection performance.

3. Proposed scheme

In this section, an efficient complex event processing method based on multiplepattern sharing is proposed for massive manufacturing event streams.

3.1. Motivation resource

There are usually many the same prefix, suffix or subpattern in the current complex event detection models during the detecting process of a complex event. However, since the existing detection methods of complex event do not consider sharing detection for many the same prefix, suffix or subpattern among them, and they can only detect a different complex event at a time separately. Detecting multiple different complex events usually needs to construct multiple different detection models of complex event, which can lead to long detection time, memory consumption, low detection efficiency due to a large number of redundant building, storing, searching and computing operations for the same prefix, suffix, or subpattern, thus influencing their whole detection efficiency.

Aiming to solve the problems above, in this paper, an efficient complex event processing method based on multipattern sharing is proposed. In our scheme, we firstly use pattern sharing technology to construct a general complex event detection model by merging all the same prefix, suffix, or subpattern that existed in single pattern complex event models by sharing technology, and then use the new detection model to quickly realize the detection of multiple different complex events at a time on the basis of the analyzing and studying the existing NFA based complex event processing methods. Our suggested method in this paper can effectively improve the current detection efficiency of complex event, thus expanding the current detection technology of complex event.

3.2. Working principle

The working principle for our scheme is that, firstly, we build up the corresponding single pattern complex event detection model based on NFA structure through given complex event detection expressions; Secondly, we use pattern sharing technology to realize the construction function for multiple pattern complex events model by merging all of the same prefix, suffix, or subpattern existed in single pattern complex event models, into a multiple pattern complex event detection model with sharing technology; At last, we use the new detection model to detect the related complex events from massive manufacturing event streams. As a result, our proposed scheme can effectively solve the problem existed in current single pattern detection methods of complex event, which can save many repetitive building, storing, searching and calculating operations, thereby improving their overall detection performance.

In our scheme, in order to quick build up the general detection model for our proposed scheme, we study the construction method of complex event detection model based on multipattern sharing by using pattern sharing technology on the basis of complex event detection model based on single pattern. In order to quick realize the detection function of our scheme, we develop the detection algorithm of complex event detection model based on multipattern sharing on the basis of the current complex event detection algorithm based on single pattern in our scheme.

3.3. Composition structure

Our suggested scheme in this paper consists of some important composition structures. **Fig. 1** is the composition structure for our suggested scheme. From the **Fig. 1,** we can see clearly that our suggested scheme mainly contains five important composition parts: read detection

expression, construct complex event detection model based on single pattern, construct complex event detection model based on multipattern sharing, match complex event detection model based on multipattern sharing and output detection results. Where read detection expression mainly executes the reading operation from detection expression of complex events. Construct complex event detection model based on single mode mainly executes the establishment function for NFA based detection models according to their given detection expression; Construct complex event detection model based on multipattern sharing primarily executes the building function for multipattern complex event detection model by traversing and merging the single pattern detection models built above; Match complex event detection model based on multipatten sharing mainly executes the detection function for the corresponding complex events with the new detection model and algorithm. Output detection results mainly executes the output function for the desired detection results when detection work finishes.

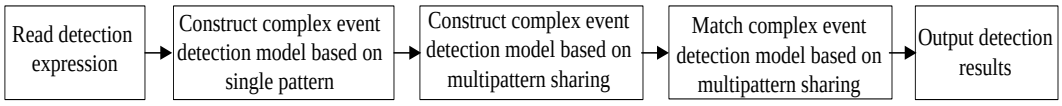


Fig. 1. Detection process of our suggested method

3.4. Realizing processing

3.4.1 Construct single pattern complex event detection model

In our scheme, we mainly construct complex event detection model based on single pattern by the use of NFA (Nondecided Finite Automaton) structure, which needs to go through the following realizing steps:

1) Get the detection expression of a complex event; 2) Construct an action for NFA based complex event model ; 3) Convert NFA into DFA(Decided Finite Automaton) and minimize its operations ; 4) Get the detection model of complex events based on minimized DFA. **Fig. 2**, **Fig. 3** and **Fig. 4** are the complex event detection model constructed based on single pattern by pattern detection expression SEQ (A, B, C, D), SEQ (A, H, F, E) and SEQ (M, E, C, A) separately. The formalization of NFA can be expressed as: $NFA = (S, \Sigma, T, S_0, A)$, where S is the set of states $\{s_1, s_2, s_3, ..., s_n\}$, $(n \geq 1)$; A is the alphabet $\{a_1, a_2, a_3, ..., a_n\}$ $(n \geq 1)$, and each element can be used as an input character ; T is a subset of image S× (Σ∨{ε}) →S; S_0 is the initial state with $S_0 \subseteq S$, non empty; A is the termination state set with A⊆S.

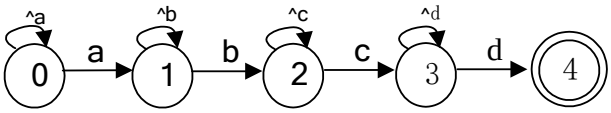


Fig. 2. Complex event detection model constructed by SEQ (A, B, C, D)

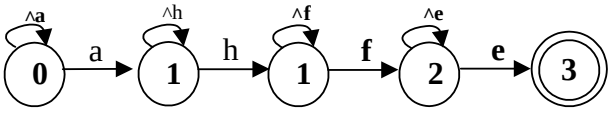


Fig. 3. Complex event detection model constructed by SEQ (A, H, F, E)

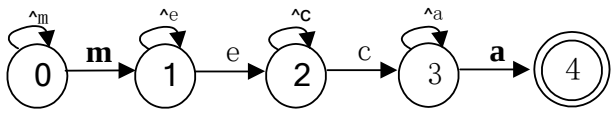


Fig. 4. Complex event detection model constructed by SEQ (M, E, C,A)

3.4.2 Construct multipattern complex event detection model

In our scheme, the constructing process for our suggested model based on multipattern sharing can be shown in **Fig. 5**. From the **Fig. 5**, we can observe that the constructing process for our proposed model includes the following three important realizing steps: traverse NFA based model, merge NFA based model and produce multipattern detection model.

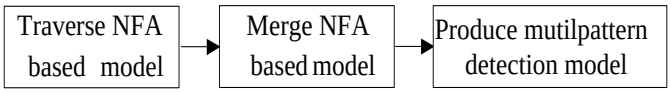


Fig. 5. The building processing for our ENFA based model

Where traverse NFA based model mainly executes the traverse operation for each NFA based detection model by using depth-first search algorithm from its starting state. Merge NFA based model mainly executes the merging operations for multipattern detection model by merging all NFA based models into a general detection model during its traversing process by sharing all of the same prefix, suffix or subpattern in them, with corresponding runtime state of automata, immigration state of automata, input condition, attributes condition, emigration condition, and so on. Product multipattern detection model mainly executes the generative function for our suggested detection model based on multipattern by merging way above.

In our scheme, in order the realize to the merging operation for the same prefix, suffix or subpattern in single pattern detection models of complex event, three important functional functions: state transition function, failure transition function and output function, are used to merge and generate our multipattern detection model during the traversing process of NFA based models in our scheme. **Fig. 6** is the merging process in our multipattern based detection model.

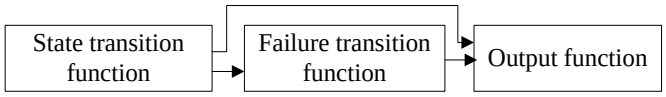


Fig. 6. The merging process for multipattern based detection model in our scheme

In **Fig. 6**, state transition function mainly executes the state transition path during traversing process for NFA based model. Failure transition function mainly executes the execution path for the next step when detection program fails to match in its traversing process. State output function mainly takes the output function when detection program needs to output detection results. **Fig. 7** is the final construction results for our scheme from complex event detection model based on single pattern by using pattern sharing technology above.

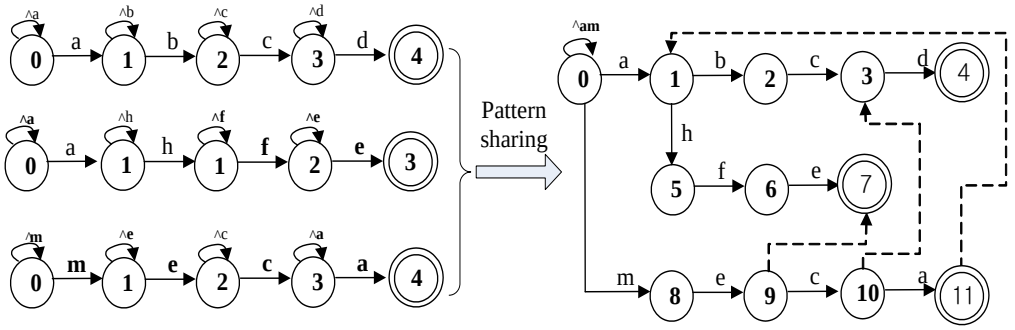


Fig.7. The final construction results for our proposed scheme

From **Fig. 7**, we can clearly observe that our proposed multipattern complex detection model includes less automata states and migration edges by sharing all of the same prefix, suffix or subpattern compared with the current single pattern complex event detection models under the same detection condition for SEQ (A, B, C, D), SEQ (A, H, F, E) and SEQ (M, E, C, A), which can greatly improve the detection performance by reducing many unnecessary runtime state transitions and migration edges for the same prefix, suffix, or subpattern in them.

3.4.3 Match multipattern complex event detection model

After constructing a multipattern complex events detection model based on NFA structure, the following content is how to quickly realize the detection for the related complex event from a massive manufacturing event stream by using the new detection model above.

In our scheme, in order to quick detect the related complex events from a massive manufacturing event stream with the new detection model, some detection steps need to go through. **Fig. 8** is the main detection process of our scheme.

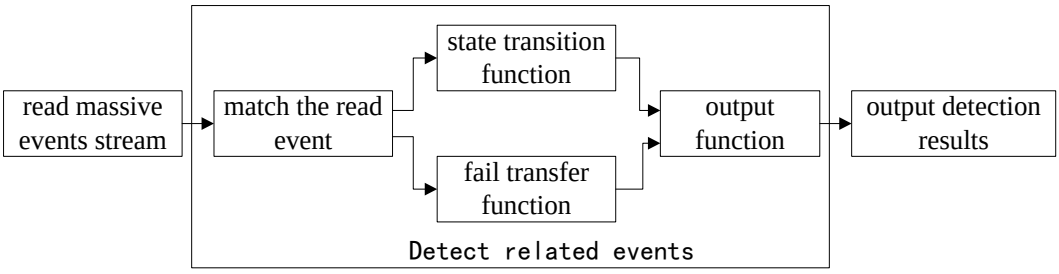


Fig. 8. Detection process of our scheme

From **Fig. 8**, we can clearly see that our proposed detection method needs to go through the following three important detection steps, read massive event stream, detect related events and output detection results, to finish its detection function. Where read massive event stream mainly executes the read operation for the related primitive events one by one from the massive manufacturing event stream. Detect related events mainly executes the matching function for the read related events from the massive manufacturing event stream, which also includes the following steps: (1) match the read event by using the new detection mode1; (2) judge whether the state transition function or fail transfer function to execute in the general

detection model according to the matching results;(3) Determine whether the output function in the general detection model is required to output the matching results according to the executed state above. Output detection results mainly executes the output operation of finial detection results when detection work finishes.

Fig. 9 is the detection process for our proposed method.

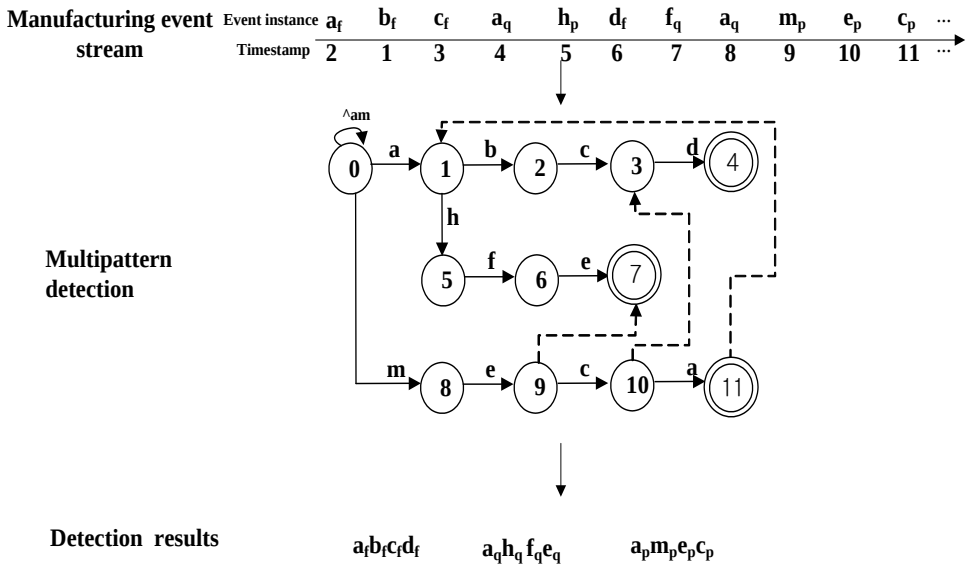


Fig. 9. The detection process for our proposed method

Fig. 9 clearly reveal the whole detection process for our proposed model. From **Fig. 9,** we can see that, in our scheme, we use multipattern sharing technology to quickly detect the related complex events from massive manufacturing event streams, which can successfully merge many the same prefix, suffix or subpattern existed in single patter complex event detection models, therefore improving the detection efficiency for current detection method.

3.5. Realizing steps of our algorithm

The realizing steps of our proposed scheme can be shown in the following **Fig. 10**, which consists of two important realizing steps:

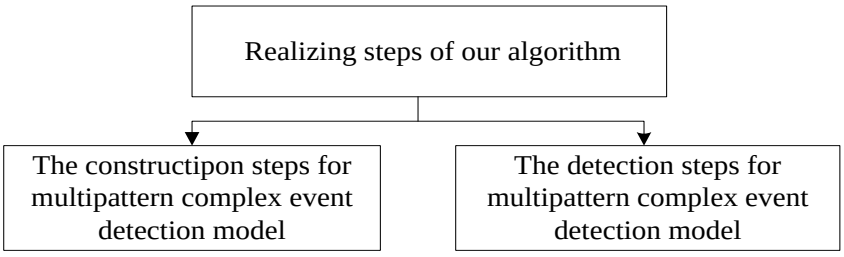


Fig.10. Realizing steps of our scheme

The first important realizing step is to construct the multipattern complex event detection model, which can be implemented by the following six realizing steps:

In step 1. read complex event detection expressions;

In step 2. build their corresponding single pattern complex event detection models based on NFA structure according to the read detection expressions of complex event above;

In step 3. construct the state transfer function of our scheme according to the built single pattern complex event detection models above;

In step 4. construct the failure transfer function of our scheme according to the execution result of the state transfer function;

In step 5. construct the output function based on the execution results of the state transfer function and the failure transfer function;

In step 6. output multipattern detection model of complex event;

In the realizing step of constructing multipattern detection model above, the main function of state transfer function is to realize the state transition path of finite automata. The primary function of failure transfer function is to point to the next state when the mismatch occurs in it. The main function of the output function is to output matching information when a pattern gets a matching.

The second important realizing step is to realize the detection function for multiple pattern detection model, which can be implemented by the following seven realizing steps:

In step 1. read the primitive event one by one from the massive manufacturing event streams;

In step 2. match the read primitive event by using our suggested multipattern complex event detection model ;

In step 3. determine whether state transition function is executed according to the matching result of read primitive event;

In step 4. determine whether failure transfer function is executed according to the executed results of state transition function;

In step 5. determine whether to execute the output function according to the executed results of state transition function and fail transfer function;

In step 6. determine whether the detection work is finished; if finish, jump step 7 to execute; otherwises, jump step 1 to execute;

In step 7. output the detection results.

In the realizing step for the detection function of multipattern detection model above, four important executed functions: read primitive event, match the read primitive event, process the matching result and output matching result, are used to complete the detection for related complex events from massive manufacturing event streams in our scheme. The output function will output related detection results when detection work finishes.

4. Experimental Results and Analysis

In this section, in order to verify the effectiveness of our proposed method above, some emulation experiments are demonstrated. Our designed experiments mainly include two part contents: build experimental environment and test the effectiveness of our proposed method respectively.

4.1. Build experimental environment

We implemented our simulation experiment on Microsoft Windows 7 operating systems, AMD A6-3420M 4 core CPU Processor, 2G memory, 500G Hard disk. In our experiment, we take the Visual C++ 6.0 tool to develop an even generator, and then use the even generator to generate different kinds of RFID manufacturing event streams by controlling some parameters.

The main setting parameters in our experiment are shown in **Table 1**.

In our experiments, we test the effectiveness of our proposed processing method in this paper. In the detection model aspects, in order to verify the effectiveness of our detection model, we select the automata states number and migration edges number as our two important comparison indicators in this paper. Where the number of automata states refers to the total number of automata states that are used to construct detection model of complex event according to the corresponding detection expression of complex event, while the number of migration edges represents the total number of migration edges that are used to build complex event model based on the corresponding detection expression of complex event.

In the detection algorithm aspects, in order to evaluate the effectiveness of our detection algorithm, we select detection time, memory consumption and event throughput as our comparison indicators respectively. Where the detection time refers to the total executed time for detecting the expected complex event from massive RFID event streams. The memory consumption refers to the total used memory consumption for detecting the desired complex event from massive RFID event streams. The event throughput refers to the detection number of the expected complex events from the massive RFID event stream in one unit time.

The comparison methods in this paper are a single pattern detection method and Zhang' method[32], where an single pattern detection method mainly uses traditional NFA technology to complete the construction and detection function for the complex event detection according to different detection expressions. Zhang' method mainly uses a series of optimized algorithms based on nondeterministic automata model to reduce its bottlenecks and realize its complex event detection after analyzing its complexity of expensive queries of complex event, while our proposed multipattern detection method in this paper mainly use multipattern sharing technology to successfully realize the quick detection of multiple different complex events with a general detection mode and algorithm, which is introduced in Section 3 of this paper.

Table 1. Main setting parameters in our experiment

Parameter name	parameter value
Sliding window size	40
Number of event types	10
Number of complex event detection expression	40
Length of complex event detection expression	3~10
scale of event stream	10^5
Number of attributes of each event	2

In our experiment, In order to reduce the randomness of testing, the average 5 testing are taken in our experiment. The experimental conditions are shown as follows.

4.2. Test the effectiveness of our detection method

4.2.1 Test the number of automata states and migration edges

In the subsection, we evaluate the effectiveness for our proposed detection model compared with single pattern detection model from automata states number and migration edges number. The experimental results are shown as follows.

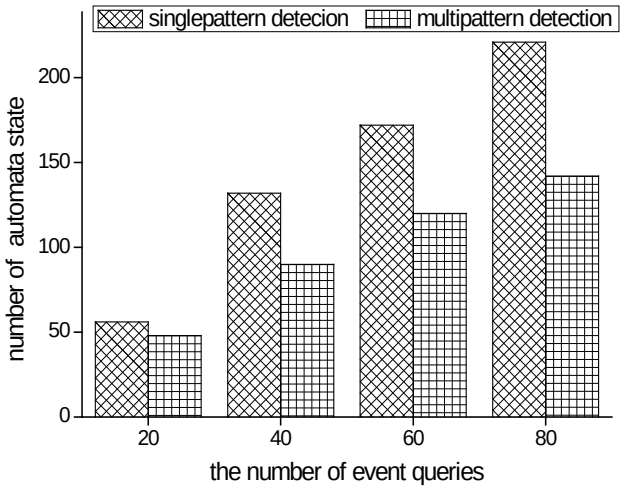


Fig. 11. Comparison result of automata states number

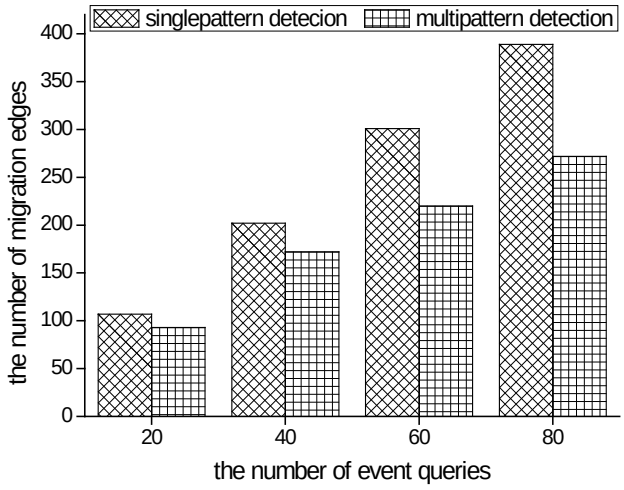


Fig. 12. Comparison result of migration edges number

Fig. 11 is the comparison result of the automata states number. **Fig. 12** is comparison result of the migration edges number. **Fig. 11** and **Fig. 12** revel that our detection model needs less automata states and migration edges compared with single pattern detection model under the same detection conditions. The reasons for that lies that, in our scheme, we mainly use multipattern sharing technology to successfully merge many the same prefix, suffix, or subpattern that existed in many single pattern detection expressions, which can eliminate lots

of redundant automata states and migration edges, therefore saving more automata states and migration edges. However, since single pattern detection model needs to construct multiple different complex event detection models according to their corresponding detection expressions due to not sharing their same prefix, suffix, or subpattern, which generates many redundant automata states and migration edges, therefore lead to more automata states and migration edges.

4.2.2 Test Detection time

In the subsection, we evaluate the effectiveness of our proposed detection algorithm from detection time. The experimental result is shown in **Fig. 13** as follows.

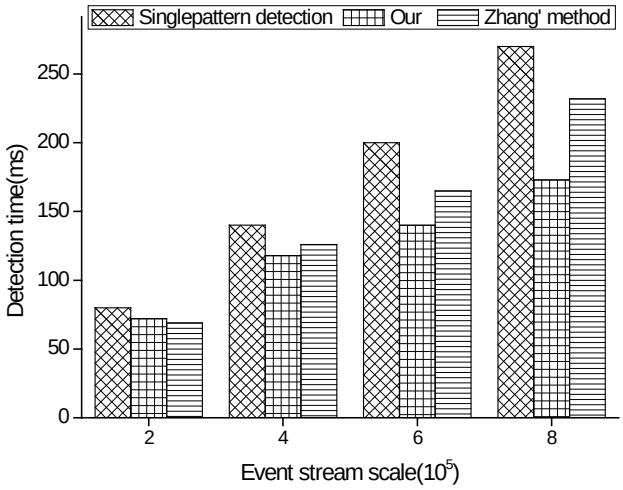


Fig. 13. Comparison result of the detection time

From **Fig. 13**, we can see clearly that our proposed algorithm shows the least detection time in three detection methods. Zhang' method followed. Single pattern detection method consumes the largest detection time. The main reasons for them are that, in our suggested approach, we use pattern sharing technology to eliminate many the same redundant prefix, suffix, or subpattern that existed in many single pattern detection models, which can reduce their many building, storing, searching and calculating operations, thus saving much detection time. Zhang' method spends less detection time lies in its use of series of optimized algorithms based on nondeterministic automata model to realize its detection function in its process of complex event. It has the similar performance as our scheme when the scale of event stream is low, but with the increase of event stream, it generates a large amount of runtime intermediate state sequences, which needs to wait for further matching computations due to its redundant intermediate processing results, therefore consuming more detection time in high event stream scale compared with our proposed method. Single pattern detection method costs the most detection time because it is unable to realize the sharing detection for many the same prefix, suffix, or subpattern during its detection process and needs to repeatedly execute the building, storing, looking-up and calculating operations for the same prefix, suffix, or subpattern, therefore consuming more detection time.

4.2.3 Test memory consumption

In the subsection, we evaluate the effectiveness of our proposed detection algorithm from memory consumption. The experimental result is shown in the following **Fig. 14**.

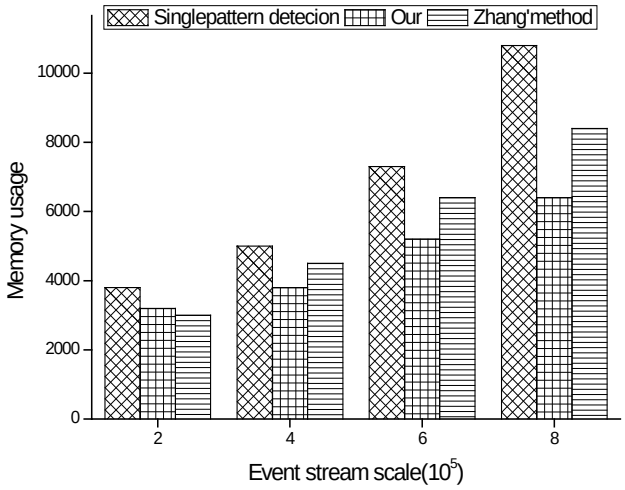


Fig. 14. Comparison result of the memory consumption

Fig. 14 reveals that our proposed algorithm in this paper has a significant improvement in saving memory consumption compared with other two methods. Zhang' method followed. Single pattern detection method shows the largest memory consumption. The main reasons for that can be explained as follows: in our scheme, a multipattern sharing technology is used to merge many the same prefix, suffix, or subpattern that existed in single pattern complex event patterns, which can reduce many redundant building, storing, looking-up and calculating operations for them, thus saving more memory usage. In Zhang' method, since it uses of series of optimized algorithms based on nondeterministic automata model to reduce its bottlenecks and realize its complex event detection in its complex event processing. It can cost less memory consumption when the scale of event stream is low, but with the increase of event stream scale, a large amount of runtime intermediate state sequences are generated, which needs to consume more memory due to its redundant intermediate processing results, therefore consuming its more memory consumption in high event stream scale. Single pattern detection method requires the most memory consumption lies that it is unable to share their the same prefix, suffix or subpattern among their detection expressions, which needs to cost much extra memory for the building, storing, looking-up and calculating operations for these same prefix, suffix, or subpattern, therefore consuming the most memory.

4.2.4.Test detection throughput

In the subsection, we evaluate the effectiveness of our proposed detection algorithm from event throughput. The experimental result is shown in the following **Fig. 15**.

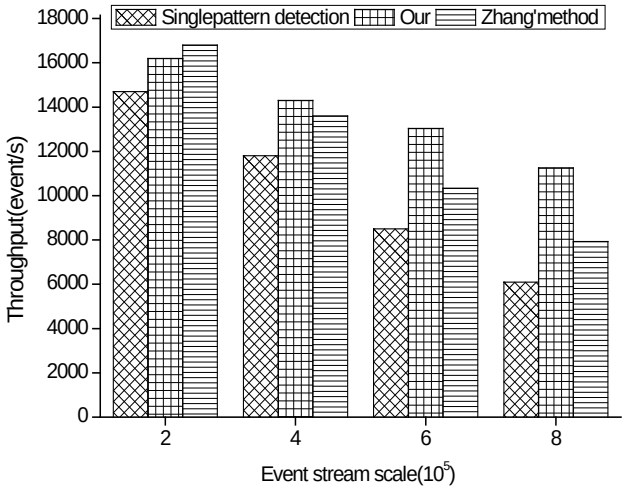


Fig. 15. Comparison result of the detection throughput

From **Fig. 15**, we can observe that our proposed method presents the good event throughput processing capacity compared with other two methods with the same event stream scale. Zhang' method followed. Single pattern detection approach presents the worst processing performance.The main reason lies in the use of multipattern sharing technology in our scheme. Specially, in our method, we use multipattern sharing technology to quickly realize the merging operation for many the same prefix, suffix, or subpattern that existed in many single pattern detection models, which can save many building, storing, looking-up and calculating operations, and then improve the detection throughput for our scheme. Zhang' method shows a better processing capability because of the uses of series of optimized algorithms based on nondeterministic automata model in its complex event processing. It presents the similar processing performance as our proposed method when the scale of event stream is very low. However, with the increase of event stream scale, it starts to generate a large amount of runtime intermediate sequences,which cannot be processed in time due to its redundant intermediate processing results, therefore leading to relatively low event throughput compared with our proposed method in high event stream scale. Single pattern detection method presents the worst processing performance because it needs to repeatedly execute the building, storing, looking-up and calculating operations for the same prefix, suffix or subpattern during their detecting process of complex event due to not pattern sharing, which needs to execute many unnecessary detection operations, therefore leading to low event detection performance.

In addition, from **Fig. 13** to **Fig. 15** above, we can also see that three detection methods show the similar detection performance in small event stream scale, but with the increasing of event stream scale, our proposed scheme is obviously superior to other two methods.

5. Conclusions

In this paper, an efficient complex event processing scheme based on multipattern sharing is presented for massive manufacturing event stream. In our scheme, we successfully use multipattern sharing technology to realize the sharing detection for many the same prefix, suffix or subpattern existed in current single pattern complex event models due to not sharing,

which can reduce lots of redundant building, storing, searching and calculating operations by taking advantage of the pattern sharing technologies. The simulation results show that our proposed scheme in this paper has a great improvement in saving the number of automata states and migration edges, reducing detection time, lowering memory access and raising event throughput compared with some general detection methods in processing massive manufacturing event streams.

References

[1] Cheng L, Wang T, Hong X, et al., "A study on the architecture of manufacturing internet of things," *International Journal of Modelling, Identification and Control*, vol.23, no.1, pp. 8-23, 2015. Article (CrossRef Link).

[2] Tao F, Cheng Y, Da Xu L, et al., "CCIoT-CMfg: cloud computing and internet of things-based cloud manufacturing service system," *IEEE Transactions on Industrial Informatics*, vol.10, no.2, pp.1435-1442, 2014. Article (CrossRef Link)

[3] Bi Z, Da Xu L, Wang C., "Internet of things for enterprise systems of modern manufacturing," *IEEE Transactions on industrial informatics*, vol.10, no.2, pp.1537-1546, 2014. Article (CrossRef Link)

[4] Del G, Eugene W, Hee J, et.al., "SASE: Complex Event Proeessing over Streams," in *Proc. of the 3rd Biennial Conference on Innovative Data Systems Researeh*, pp.407-411, Dec, 2007. Article (CrossRef Link)

[5] Demers A, Gehlke J, Panda B, et al., "Cayuga: A General Purpose Event Monitoring System. Comell University," in *Proc. of the 3rd Biennial Conference on Innovative Data Systems Research*, Asilomar, USA, pp.412-422. 2007. Article (CrossRef Link)

[6] Esper Introduction, http://esper.codehaus.org. Article (CrossRef Link)

[7] Garg V., "Estream: an Integration of Event and Stream Proeessing," Master Thesis, *University of Texasat Arlington*, 2005. Article (CrossRef Link)

[8] Cybenko G, Berk H. "Proeess Query System," *Computer*, vol.40, pp.62-70, 2007. Article (CrossRef Link)

[9] Viatkin V, Nakano K, Hayashi T., "Optimized Processing of Complex Events in Discrete Control Systems using Binary Decision Diagrams," *IFAC Proceedings Volumes*, vol.30, no.3, pp. 403-408, 1997. Article (CrossRef Link)

[10] Bai L, Lao S, Smeaton A. F, et al., "Semantic analysis of field sports video using apetrinet of audio-visual concepts," *Computer*, vol.52, no.7, pp.808-823, 2009. Article (CrossRef Link).

[11] Xiangwei Sun, Rong Chen, Zhenjun Du. "Composite Event Detection Based on Automata," in *Proc. of 2009 IEEE International Conference on Intelligent Human-Machine Systems and Cybernetics*, pp. 160-163, Aug 26-27, 2009. http://dx.doi.org/10.1109/IHMSC.2009.48

[12] Mei Y, Madden S., "ZStream: A cost-based query processor for adaptively detecting composite events," in *Proc. of the 2009 SIGMOD*. USA, pp.193-206, June, 2009. Article (CrossRef Link).

[13] Jin X, Lee X, Kong N. "Efficient complex event processing over RFID data stream," in *Proc. of the Seventh IEEE/ACIS International Conference on, Portland*, May, pp.75–81, 2008. Article (CrossRef Link)

[14] Wang F, Liu S, Liu P., "Bridging physical and virtual worlds:complex event processing for RFID data streams," in *Proc. of the 10th International Conference on Extending Database Technology, EDBT' Munich, Germany*, March, pp.588–607, 2006. Article (CrossRef Link).

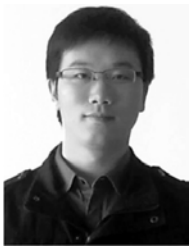
[15] Li Y, Wang J, Feng L., "Accelerating sequence event detection through condensed composition," in *Proc. of the 5th International Conferenceon Ubiquitous Information Technologies & Applications*, Sanya, pp.6-10, 2010. Article (CrossRef Link)

[16] Cao J, Wei X, Liu Y Q, et al., "LogCEP-Complex event processing based on pushdown automaton," *International Journal of Hybrid Information Technology*, vol.7, no.6, pp.71-82, 2014. Article (CrossRef Link)

[17] Eugene W, Yanlei D, Shariq R., "High-performance Complex Event Proeessing over Streams," in *Proc. of the SIGMOD conf.*, Chicago USA, pp.407-418, June 2006. Article (CrossRef Link)
[18] Zhang X L, Wang Y H, Zhang X M. "Complex Event Processing over Distributed Uncertain Event Streams," in *Proc. of International Conference on Computer Science and Service System*, pp.357-361, July, 2014. Article (CrossRef Link)
[19] Kyoung S B, Myung H Y, Byoung Y L, et al., "Efficient Complex Event Processing over RFID Streams," *International Journal of Distributed Sensor Networks*. vol.2012, pp.1-9, 2012. Article (CrossRef Link)
[20] Wang J H, Cheng L L, Liu J., "A Complex Event Detection Method for Multi-probability RFID Event Stream," *Journal of Software*. vol.9, no.24, pp.834-840, 2014. Article (CrossRef Link)
[21] Peng S L, Liu H X, Guo X L, et al., "State-based event detection optimization for Complex Event Processing," *Sensors and Transducers*, vol.164, no.2, pp.242-248, 2014. Article (CrossRef Link)
[22] Wang Y H,Cao K, Zhang X M. "Complex event processing over distributed probabilistic event streams," *Computers and Mathematics with Applications*, vol.66 no.10, pp.1808-1821, 2013. Article (CrossRef Link)
[23] Peng S L, Li Z H, Cheng, et al. "Complex event processing over live archived data streams," *journal of computers*.vol.35, nol.3, pp.540-554, 2012. Article (CrossRef Link)
[24] Moon M, Kim S, Yeom K, and Choi H, "Framework for extending RFID events with business rule," in *Proc. of the International Conference on Database Systems for Advanced Applications*, pp.955–961. 2007. Article (CrossRef Link)
[25] Kawashima H, Kitagawa H, and Li X, "Complex event processing over uncertain data streams," in *Proc. of the International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, pp. 521–526, 2010. Article (CrossRef Link)
[26] Liu Y and Wang D, "Complex event processing engine for large volume of RFID data," in *Proc. of the 2nd Inter-national Workshop on Education Technology and Computer Science*, pp. 429–432, March, 2010. Article (CrossRef Link).
[27] Wang J H, Wang T, Cheng L L, Lu L L., "An Efficient Complex Event Processing Algorithm based on INFA-HTS for Out-of-order RFID Event Streams," *KSII Transactions on Internet and Information Systems*, vol.10, no.9, pp.4307-4325, 2016. Article (CrossRef Link).
[28] Xiao F, Zhan C, Lai H, et al., "New parallel processing strategies in complex event processing systems with data streams," in *Proc. of the International Journal of Distributed Sensor Networks*, vol.13, no.8, 2017. Article (CrossRef Link)
[29] Yuan L, Xu D, Ge G, et al., "Study on distributed complex event processing in Internet of Things based on query plan," in *Proc. of the IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems*, pp.666-670, Shenyang, China , June, 2015. Article (CrossRef Link)
[30] Xiao F. "An Intelligent Complex Event Processing with D Numbers under Fuzzy Environment," *Mathematical Problems in Engineering*, vol.2016, no.1, pp.1-10, 2016. Article (CrossRef Link)
[31] Abbasnejad I, Sridharan S, Denman S, et al. "Complex Event Detection Using Joint Max Margin and Semantic Features," in *Proc. of IEEE International Conference on Digital Image Computing: Techniques and Applications*, pp.1-8, 2016. Article (CrossRef Link)
[32] Zhang H, Diao Y, and Immerman N, "On complexity and optimization of expensive queries in complex event processing," in *Proc.of ACM SIGMOD*, pp. 217–228, Snowbird, USA, 2014. Article (CrossRef Link)

JianHua Wang was born on February 6, 1982 in Guangdong, China. He received his B.S degree in Electronic Information Science and Technology from Shaoguan University, Guangdong, China, in 2006. He received his Ph.D degree in Control Science and Engineering at Guangdong University of Technology, Guangdong, China, in 2015. Currently he is a teacher of college of electronic engineering, south china agricultural university, Guangzhou, China. His research interests include wireless video transmission, cyber-physical systems and IoT.

Yubin Lan was born on July 14, 1961, Doctor, Director and Chief Scientist of International Lab of Agricultural Aviation Pesticide Spraying Technology; He is professor and doctoral supervisor of South China Agricultural University, senior researcher of USDA, and chairman of CIGR Precision Aerial. He is employed as "Communication Specialist of the National Agricultural Aviation Technology" of CAST, and Deputy Director of AOPA Specialist Committee. He has engaged in application and development of precision agricultural aviation, aviation pesticide spraying technology and aerial remote sensing technique.



Shilei Lu was born on January 17, 1984. He received his B.S. degree in electrical automation engineering from Northeastern University, Qinhuangdao, China, in 2006 and received his PHD Degree from Sun Yat-Sen University in 2013. Currently he is a teacher of college of electronic engineering, south china agricultural university, Guangzhou, China. His research interests include Agricultural Internet of things and intelligent optimization, RFID network security based on IOT.

LiangLun Cheng was born on August 22, 1964 in Hubei. He received his M.S and Ph.D degrees from Huazhong University of Science and Technology, Hubei, China in 1992 and Chinese academy of Sciences Jilin, china in 1999 respectively. He is a Prof and doctoral supervisor of Guangdong University of Technology. His research interests include RFID and WSN, IoT and CPS, production equipment and automation of the production process, embedded system, the complex system modeling and its optimization control, software of automation and information, etc.