Computers and Electrical Engineering 108 (2023) 108671

Contents lists available at ScienceDirect

Computers and Electrical Engineering

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

Policy conflict detection approach for decision-making in intelligent industrial Internet of Things

Pradyumna Kumar Tripathy [a], Mohammad Shabaz [b,*], Abdelhamid Zaidi [c], Ismail Keshta [d], Uttam Sharma [e], Mukesh Soni [f], Anurag Vijay Agrawal [g], Renato R. Maaliw III [h], D.P. Sharma [i]

[a] Department of Computer Science and Engineering, Silicon Institute of Technology, Bhubaneswar, Odisha
[b] Model Institute of Engineering and Technology, Jammu, J&K, India
[c] Department of mathematics, College of Science, Qassim University, P.O.Box 6644, Buraydah 51452, Saudi Arabia
[d] Computer Science and Information Systems Department, College of Applied Sciences, AlMaarefa University, Riyadh, Saudi Arabia
[e] Department of computer science and engineering, Gautam Buddha University, Greater Noida, India
[f] Department of CSE, University Centre for Research & Development Chandigarh University, Mohali, Punjab-140413, India
[g] Consultant, E & ICT Academy, Department of Electronics and Communication Engineering, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand, India
[h] College of Engineering, Southern Luzon State University, Lucban, Quezon, Philippines
[i] School of Computing and Information Technology, Manipal University Jaipur, India

ARTICLE INFO

This paper is for CAEE special section VSI-apsi. Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr Gupta Deepak.

Keywords:
Internet of Things
Intelligent industrial Internet of Things
Artificial intelligent
Rule conflict detection
Policy conflict
Decision making

ABSTRACT

Improving Industrial Internet of Things (IIOT), device flexibility and lowering maintenance costs are significant problems. However, as the scale of the Intelligent Industrial Internet of Things (IIIOTs) system expands, the interactions between the rules get more sophisticated, potentially leading to rule discrepancies. The algorithm for Formal Rule Conflict Detection (FRCD) is created, and a thorough explanation of the procedure is given in this paper. Two IIIOT systems were used in experiments and the results were compared with three existing standard IIIOT rule conflict detection techniques. They include policy conflict detection systems based on Web semantics (Semantic Web-Based Policy Interaction Detection with Rules, (SPIDER)), conflict detection methods based on Users, Triggers, Environment entities, and Actuators (UTEA), and semiformal conflict detection methods (Identifying Requirements Interactions using Semiformal (IRIS)). The experimental results show that the FRCD rule conflict detection method is superior.

1. Introduction

The Internet of Things refers to the Internet between objects and objects. Through wireless sensing technology, it uses sensors to obtain information about objects and the environment. It realizes information transmission and resource sharing between physical devices and between physical devices and networks [1]. Fig. 1 is a typical IIOT system architecture, mainly divided into three parts: external elements, network layer, and control system. Exterior features include sensors and devices. Sensors are the source of

This paper was recommended for publication by Associate Editor Dr Gupta Deepak.
* Corresponding author.
E-mail address: bhatsab4@gmail.com (M. Shabaz).

https://doi.org/10.1016/j.compeleceng.2023.108671
Received 15 January 2023; Received in revised form 3 March 2023; Accepted 8 March 2023

information flow and can collect physical quantities such as temperature, humidity, light intensity, and pressure. Devices include programmable hardware and are destinations for information flow. IoT sensors and actuators are part of an IoT subsystem, which is controlled by one or more controllers. It might also include some intermediary technology, such data aggregators (aggregating data from multiple sensors with the same functionality). The network layer completes the information transmission and realizes the connection between external elements and the control system [2]. Because of network-to-network communication, the Internet is only feasible. The "network layer" of the Internet communications process establishes connections by sending data packets back and forth across distinct networks [3]. Everything that has to do with inter-network connections happens at the network layer. This includes identifying the paths that data packets will go, detecting whether a server on another network is operational, and addressing and receiving IP packets from other networks. The user is not a part of the targeted IIoT system. This is done to explicitly separate usage relationships from system-internal interactions. Fig. 2 is an architecture that uses rule reasoning as the core of the control system, which includes an interactive processing module and a rule reasoning module. Therefore, when the knowledge is deployed in the inference engine, the inference engine can perform logical reasoning on the state data of the external input elements according to the knowledge [3]. The process of deducing what we believe to be true from what we know to be true is called inference. For example, Fig. 3 shows two typical rule conflicts in the Internet of Things. Fig. 3(a) is a case of rule conflict. The user writes two rules. Rule R1: When the temperature is less than 25°C, turn on the electric heater to raise the temperature to 27°C. Rule R2: When the temperature exceeds 20°C, turn on the air conditioner to cool down to 15°C. At this time, the ambient temperature is 22°C, and both rules are triggered. Unfortunately, electric heating and air conditioning have opposite effects on the ambient temperature, resulting in adverse impact conflicts.

Fig. 3(b) is another case of rule conflict. The user writes three rules. Rule R3: When the light is on, close the curtains. Rule R4: When the lights are on, open the curtains. Rule R5: When the light intensity is less than 3000lm, turn on the light. At this time, the ambient light intensity is 2000lm, and all three rules are triggered. Affected by rules R3 and R4, the curtains are in the state of continuous opening and closing, which results in a conflict of execution. The conflicts between the rules are shown in Fig. 3(a) and Fig. 3(b), because the existing Internet of Things rule conflict classification is not fine enough, the current conflict detection methods may not be able to detect these two rules. Conflicts result in the problem of missed detection of rule conflicts.

Aiming at the problem of rule conflicts in the Internet of Things system, some scholars have researched it. Author [4] proposed a requirement interaction taxonomy used to classify and identify the requirement interaction in software systems. Author [5] analyzed the ontology model of the intelligent home system to realize knowledge reuse and contextual semantic modeling. They proposed a semantic web-based policy interaction detection with rules (SPIDER) method to detect intelligent home systems. Rule conflicts in home services provide functional support for the ontology editing tool Protégé [6] and the rule engine tool jess [7]. However, the above two methods only consider discrete system states when classifying rule conflicts. Author [8] proposed a conflict detection method based on users, triggers, environmental entities, and actuators (user triggers, environment entities, and actuators, UTEA) based on the analysis of the status quo of intelligent homes.

The main contributions of this paper are as follows:

1) By researching and analyzing the rules of the existing Internet of Things system, the rule conflicts in the current Internet of Things system are subdivided into seven categories, namely, execution coverage conflicts, execution conflicts, harmful impact conflicts, complete resource and direct conflicts, Ignore explicit circular dependency conflicts, explicit circular dependency conflicts and indirect circular dependency conflicts.
2) Based on the rule conflict classification of the Internet of Things system, different rule conflicts are formally expressed so that the detection of rows can be automated, and other rule conflicts can be well distinguished. To a certain extent, it can improve the accuracy of rule conflict detection.
3) This paper designs and implements a prototype system for rule conflict detection and conducts experimental verification in two IIOT systems. The experimental results show that the method in this paper is better than the other three in the F1 value of rule conflict detection in the IIOT system. The conflict detection method is better.

1.1. *Organization of the paper*

The present section introduces the concept of philosophy and recent contributions of Policy Conflict Detection Approach over

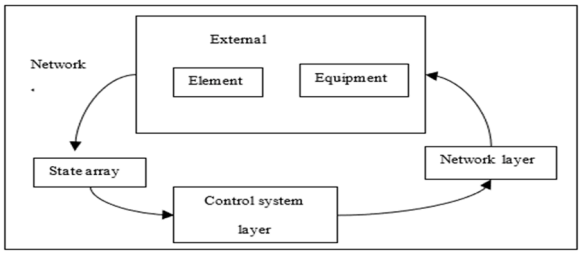


Fig. 1. Typical IIOT system architecture

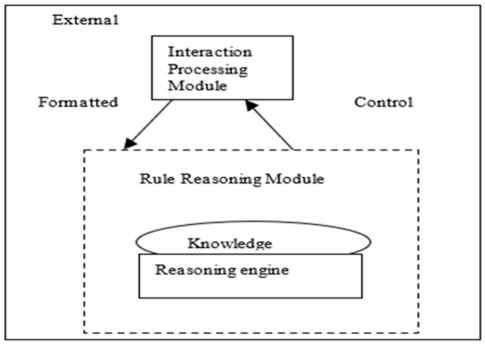
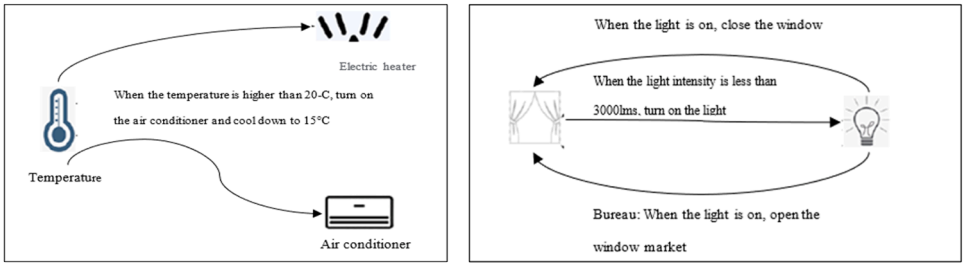


Fig. 2. Architecture of rule reasoning control system



(a) Rule Conflict Case 1 **(b) Rule Conflict Case 2**

Fig. 3. Rule Conflict Case

Intelligent Industrial Internet of Things for decision-making. Section 2 describes the Related Work with results like Rule Authorization Access and Rule Integrity, Rule Conflict Detection. Section 3 describes the methods with results like formal analysis of rules for IIOT systems, interaction relationship of rules, Conflict Types of rules, conflict detection methods based on formal rules, rule preprocessing, analysis of rule interaction relationship, rule conflict detection, example of rule conflict detection. The Penalty making section describes the Experiment and Result Analysis with results like Research Questions, Subjects, Experimental Environment and Parameter Settings, Evaluation Indicators, Experimental Results and Analysis. The last section summarizes the findings as a conclusion with regard to the present work.

2. Related work

2.1. Rule authorization access and rule integrity

Author [9] aimed at the authorization problem caused by RIF [10] rule inference, proposed the application graph marking algorithm to solve the problem of RDF [11] tuple security signature inconsistency caused by rule inference. Author [12] proposed an authorization rule specification language model to solve the problem of simultaneous acceptance and rejection of the same service by the rules in the authorization model. Authors [13] analyzed the role-based rule access control strategy. They believed that the multiterminal decision graph is a scalable access control strategy solution and realizes the verification method of the rule access control strategy. Author [14] proposed a formal rule checker to ensure the behavioral security of controllers and actuators through controller security policies. Author [15] proposed a platform-centric IIOT centralized auditing method for IIOT security audit logs scattered on various devices and cannot be used to reconstruct security transaction workflows. Efficient automated testing of device APIs and device APIs to generate audit logs of system activity, including malicious behavior. Author [16] aimed at the problem that wrong device control affects the correctness of the system. They proposed an end-to-end linear mixed automaton model to assist non-professional Internet of Things users in checking the rules and ensuring the availability of networked systems. Authors [17] believed that the method based on the rule graph could solve the problem of data inconsistency and use the rule graph to describe the hierarchical structure of the rules and dynamically evaluate the consistency of the data. Author [18] developed a static technology to solve the problem that users often make mistakes in the triggering part of writing rules, which can automatically generate the correct triggering conditions of regulation according to the actions written by users. Author [19] extracted the Internet of Things descriptors to standardize the authorities to solve the problem of missing data in rule calls and improve the rules' quality. Both traditional industrial control systems and modern Industrial IoT make heavy use of intrusion detection systems [20]. A system for IICS anomaly detection based on deep learning was presented in 2019 by Jiang et al. [21]. Information that can be used for learning and verification is contained in TCP/IP data packets. Sunny et al. [22] used the new UNSWNB15 data set to train multilayer deep neural networks and

used Bi-LSTMGRNN to predict Industrial IoT hazards in 2020. In order to protect the Industrial Internet of Things against DDoS attacks and other cyberattacks, namely, cyberattacks against SCADA systems, Chen et al. [23] created a network intrusion detection system based on a CNN-based data gathering and monitoring control system in 2020. In order to detect such cyber-attacks, Patil and Sankpal [24] researched power theft attacks on intelligent grids and created a deep learning-based intrusion detection system.

2.2. Rule conflict detection

Yuan et al. introduced the Deep Fed federated deep learning technique for identifying and mitigating cyber threats to the dispersed Industrial Internet of Things in 2021 [25]. However, today's high-speed, large-capacity, and complex multidimensional data are beyond the capability of the aforementioned approaches. A lengthy training procedure is usually necessary for Industrial IoT data. As a result, the accuracy must be enhanced. Suwant Tolai and Polprasert [26] presented an intrusion detection system based on reinforcement learning for monitoring and analyzing sensor networks in 2020, in contrast to adaptive machine learning-based intrusion detection systems and cluster hybrid intrusion detection systems. Zhou et al. [27] presented a context-adaptive intrusion detection system in 2020 to improve the detection accuracy of increasingly evolving complex network assaults. This system employs a large number of independent deep reinforcement learning agents distributed throughout the network. Wu and Feng [28] reported a partially observable Markov decision process based on a model-free reinforcement learning algorithm in 2017 to identify online network risks. Wang et al. [29] introduced an adversarial multiagent reinforcement learning model for intrusion detection systems in 2020. However, there is still potential for growth and enhancement in terms of deep reinforcement learning-based intrusion detection system training efficacy and precision.

3. Methods

3.1. Formal analysis of rules for IIOT systems

To express the rules in the Internet of Things system more clearly and distinguish different types of rule conflicts, this paper gives the corresponding formal structure for the rules in the Internet of Things system. Internet of Things practice involves five components: control, subject, action, trigger, condition, power, and symbols. The specific structure is shown in Fig. 4:

1) The control subject sub comprises identification id, subject type, occupancy flag, mon, subject attribute, and attribute value. Among them, the type and attribute are string type, mon= {0, 1}, and the value is numeric. Therefore, the control subject can be expressed as sub= {id, class, mon, attribute, value}.
2) The action is composed of the control subject sub who executes the action, the attribute affected by the action, the action relational operator op, and the attribute value of the operation. Where oq={<, =, >, ≤, ≥, !=}. Actions can be expressed as action= {sub, attribute, op, value}. The set of actions can be expressed as actions= {action (i) | $0 \le i < n$, $n \in N$}.
3) The trigger condition is composed of the type of the control subject; the constraint attribute, the constraint relational operator op, and the constraint attribute value. For example, style refers to the element type in the control subject sub in the above rule components. Therefore, the trigger condition can be expressed as condition = {type, attribute, oq, value}.
4) The set of trigger conditions can be expressed as conditions = {condition (i) | $0 \le i < n$, $n \in N$}.
5) The rule is composed of id, trigger conditions, and actions.
6) A rule can be expressed as rule = {id, condition, action}. A collection of rules can be described as rules = {rule(i)|$0 \le i < n$, $n \in N$}.
7) To express the continuous system state, an operator # is defined to transform the discrete system state value into an interval. Details are as follows:

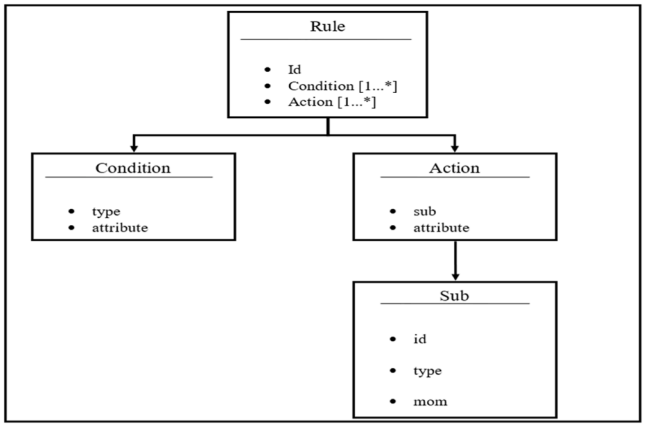


Fig. 4. Rule structure

$$oq \ \backslash \#\text{value} \begin{cases} \text{value},\ oq = '=', \\ (\ \text{value}\ , +\infty), oq = '>', \\ (-\infty,\ \text{value}\ ), oq = '<', \\ [\ \text{value},\ +\infty), oq = '\geq', \\ (-\infty,\ \text{value}\ ], oq = '\leq', \\ (-\infty,\ \text{value}\ ) \cup (\ \text{value}\ , +\infty), oq = '\neq''. \end{cases}$$

Among them, the value is a numerical value. After adding the symbol #, the numerical value is operated with the relation symbol oq, and the numerical value is converted into a continuous interval. Where -∞ stands for negative infinity, +∞ stands for positive infinity, () is an open interval, [a] is a closed interval, (a] is an open and fast interval, and [) is a short and open interval.

The following rule interaction relationship can be clearly expressed through the above rule structure representation.

3.2. Interaction relationship of rules

The interaction relationships of the eight directions are shown in Table 1:

1) CC indicates that two rules can be triggered in the same system scenario, which is called a compatible trigger condition;
2) SSS indicates that the two rules control the same non-exclusive type of subject, which is called holding the same non-exclusive issue;
3) SMS indicates that the two rules control the same exclusive subject, which is called controlling the same entire body;
4) SA indicates that the two rules have the same control action, which is called the same control action;
5) DA indicates that the two rules have opposite control actions, called opposite control actions;
6) MV indicates that the influence of two rules on environmental attributes is mutually exclusive, which is called mutually exclusive influence value;
7) RC (S_i, S_j) means that the triggering condition of the rule S_i depends on the action of the law S_j, which is called the action of the triggering state of the law S_i depending on the direction S_j;
8) OR (S_i, S_j) means that the triggering condition of the rule S_i depends on the reverse action of the law S_j, which is called the reverse action of the triggering state of the government S_i depending on the direction S_j.

Through the summary of the interaction relationship of the above rules, a symbolic representation is provided for the formal expression of the following rule conflict types.

3.3. Conflict types of rules

Because of the interactive relationship between the rules, conflicts between the administrations are generated. To describe the rule conflicts more clearly, this paper divides the rule conflicts in the current Internet of Things system into seven categories by investigating and analyzing the rule interaction relationship in the Internet of Things system, namely: execution coverage conflict, execution conflict, harmful impact conflict, exclusive resource conflict, direct ignore dependency conflict, direct circular dependency conflicts, and indirect circular dependency conflicts. The types of rule conflicts and their formal expressions are shown in Table 2:

Determining the scope of a conflicting regulation is the main goal of interpretation. Every dispute resolution system must delineate the borderline between the situations in which the conflict rule on contracts prescribes the applicable law from those in which the conflict rule on torts applies. The operation of many subsystems in the same shared physical environment is a significant contributor to dependency and, consequently, conflicts. Dependency conflicts can come in many different shapes and sizes.

The symmetric relationship in Table 2 means that the order of expression of the two rules is exchanged, and the semantics of the expression remains unchanged. The asymmetric relationship implies that the two rules exchange the expression order and the semantics of the expression change. For example, the rule S_i depends on the direction S_j, and the rule S_j depends on the direction S_i. After exchanging the expression order of the rules, the expressed semantics are different, which belongs to the asymmetric relationship.

Table 1
Rule Interaction Relationship

Interaction Name	Symbolic Representation
Compatible Trigger Condition	CC(ComCon)
Control same non-exclusive subject	SSS(SamShareSub)
Control same exclusive subject	SMS(SamMonSub)
Same Control Action	SA(SamAct)
Opposite control action	DA(DifAct)
Mutual exclusion influence value	MV(MutVal)
The trigger condition of rule S_j depends on the action of rule S_j	RC (S_i, Sj) (RelyCon)
The trigger condition of rule S_j depends on the reverse action of rule Sj	OR(Si,Sj)(OppRely)

Table 2
Types of Rule Conflicts

Conflict Type	Formal Expression	Relation
Execution Override Conflict	CC ∩SSS∩SA	Asymmetric
Implementation Conflict	CC∩SSS∩DA	Symmetry
Negative Impact Conflict	CC∩MV	Symmetry
Exclusive Resource Conflict	CC∩SMS	Asymmetric
Just Ignore Dependency Conflicts	OR (S_i,S_j)	Symmetry
Direct Circular Dependency Conflict	RC (S_i, S_j)∩RC S_i, S_i)	Symmetry

After the above summary of rule conflict types and formal definitions, the characteristics of rule conflicts in the IIOT intelligent system are summarized, and these rule conflicts can be distinguished. These conflicts can be detected relatively easily

3.4. Conflict detection method based on formal rules

The flow chart of the rule conflict detection method is shown in Fig. 5. The process of this method mainly includes two parts, which are rule preprocessing and rule conflict calculation. The input of the rule conflict detection method is the existing rule base and the rules to be detected. The current rule base refers to the rule set before the rule conflict detection. The rules to be checked refer to the powers that need to check whether there is a conflict with the existing rule base.

3.4.1. Rule preprocessing

The purpose of rule decomposition is to simplify the rules containing complex logic, which can facilitate subsequent rule conflict detection. In this paper, the disjunctive paradigm is used to decompose the regulations. For example, rule S_1 is transformed into S_2 through the disjunctive paradigm, and S_2 can be expressed as two rules, S_3 and S_4, whose trigger condition only contains a logic relation. This process decomposes rule S_1, which has and OR logic relations, into statutes S_3 and S_4, that only contain AND logic relations.

$$S_1 : (d_1 \cup d_2) \cap d_3 \Rightarrow b_1, b_2, \ldots, b_n$$
$$S_2 : (d_1 \cap d_3) \cup (d_2 \cap d_3) \Rightarrow b_1, b_2, \ldots, b_n$$
$$S_3 : d_1 \cap d_3 \Rightarrow b_1, b_2, \ldots, b_n$$
$$S_4 : d_2 \cap d_3 \Rightarrow b_1, b_2, \ldots, b_n$$

The rule decomposition step first inputs the rule element library and rule elements after the above disjunctive paradigm decomposition. Finally, the output decomposed rule base and the rules to be detected.

3.4.2. Analysis of rule interaction relationship

The rule interaction analysis is shown in Algorithm 1. The algorithm input rules S_i and S_j and output rule relation re. Lines 2 and 3 of the algorithms traverse the constraint conditions and action part of S_i and S_j. Lines 4 ~ 27 obtain the formalized elements of the rules and set the flag bit of the rule relationship variable according to the rule interaction relationship defined in Table 1—line 27 outputs the variable, which stores the interaction relationship between two rules. Finally, the interaction relationship between the rule to be detected and all rules in the rule base is calculated. The symbol abbreviation in Table 1 represents the name of the rule interaction relationship, and some symbols depend on Fig. 4.

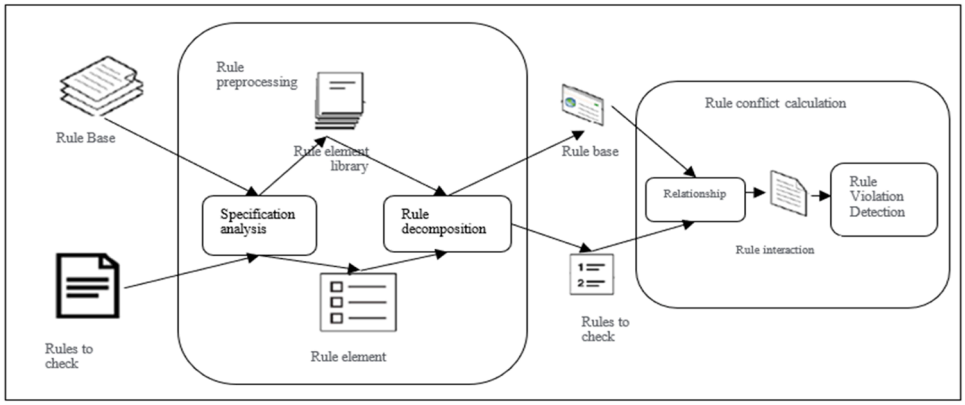


Fig. 5. Flowchart of rule conflict detection method

Algorithm 1
Rule Interaction Analysis Algorithm.

Input: Rule S_i, Rule S_j Re;
Output: Rule Interaction Relation Re.

1. Relation(s_i, s_j);
2. For ac∈ s_i.conditions, aa∈ s_i. Actions
3. For bc∈ s_j.conditions, ba∈ s_j.actions
4. If! (2 rule conditions, part type and the attribute is the same, and their oq #value has no intersection)) then
   a. Re. Cc = 1;
   b. Nd if
5. If (two rule action parts control the same sub, and they are not exclusive devices) then
6. Re. Sss = 1;
7. Else if (the two rule action parts control the same sub, and they are exclusive devices) then
   a. Re. Sms = 1;
8. End if
9. End for
10. If (the type of action part of the two rules is the same and their oq #values are containment relationships) then
    a. Re.sa = 1;
11. End if
12. If (the type of action part of the two rules is the same and their op#values do not intersect) then
    a. Re.da = 1;
13. End if
14. If (the action parts of the two rules control different sub and their oq#value have no intersection); then
    a. Re. Mv = 1;
15. Ends if
16. If (the conditional part of a rule value ∈ another, oq#value and the condition part of a rule is the same as the attribute of the action part) then
    a. Re.rc = 1;
17. Ends if
18. If (value ∈ of the condition part of a rule, is the complement of the Oq#value in the action part of another rule, and the attribute in the condition part and the action part is the same) then
    a. Re.or = 1;
19. Ends if
20. Ends for
21. Return Re.

3.4.3. Rule conflict detection

After obtaining the interaction relationship between the rules, the next step is to detect the conflict between the powers. The following steps can analyze the conflict detection between the rules. First, the dependency relationship between the law to be seen and all rules in the rule base is calculated. Secondly, the interaction relationship between the two practices currently participating in the detection is obtained. Then, match the rule conflict type—finally, output rule conflict detection information. Rule conflict detection is shown in Algorithm 2. To simplify the expression, the name of the rule interaction relationship is represented by the symbol abbreviation in Table 1. The algorithm inputs a rule based RDB and rule S_i to be detected and outputs rule conflict detection information. Line 2 of Algorithm 2 defines the variable rely, a MAP data type. Its key is the rule id, and its value is a queue composed of other rule ids

Algorithm 2
Rule Conflict Detection Algorithm.

Input: Rule Base RDB, Rule S_i;
Output: Conflicting.

1. Detect(S_i, RDB);
2. Define the MAP type variable rely
3. For IA ∈ RDB
4. Define the queue type variable rely;
5. For IB ∈ RDB
6. RAB = relation(IA, IB);
7. If IA.id != IB.id AND RAB.RC;
8. Then
9. IB deposits to rely, S_i. It is used as a key, and rely q is stored in rely m as a value;
10. End if
11. End for
12. End for
13. For S_i ∈ RDB
14. Re = relation(S_i, S_j);
15. Conflict info = match conflict(re, relym);
16. If conflict info != φ THEN
17. RETURN conflict info;
18. End if
19. End for
20. Return “No Conflict”.

on which the current rule depends. It is used to store the dependency information between the rules. No. 3-Line 11 traverses all rules in the RDB, uses the id of this rule as the key, and stores the ids of all rules that directly depend on it as values in the dependent variable. Lines 12∼18 traverse the rules S_j in the RDB, where S_i, S_j are used as the input of the function relation () and get the relation re of S_i, Sj, re and rely are used as the input of the function match Conflict (), and the rule conflict information is matched according to the rule conflict types in Table 2. Finally, it is calculated whether there is any conflict between the rule to be detected and the rules in the rule base, and if there is a conflict, the specific conflict type is output. Among them, Lines 6∼8 call the rule interaction relation analysis function relation () of Algorithm 1.

3.4.4. Example of rule conflict detection

Through the implementation of the above algorithm, the rule conflicts in Fig. 3(a) and Fig. 3(b) can be detected under any circumstances. In Fig. 3(a), the two rules R1 and R2 are, respectively, obtained after rule preprocessing: conditionS$_1$ = {room [Environment], temperature, <, 25}; actionS$_1$ = {heater, temperature, >, 27}; conditionS$_2$ = {room [Environment], temperature>, 20}; actionS$_2$ = {air_conditioner, temperature, <, 15}.

Rules S$_1$and S$_2$ are analyzed through the rule interaction relationship. For example, the room [environment] and temperature in condition S$_1$ are equal to the room [environment] and temperature in conditionS$_2$, but "<25" in conditionS$_1$ and ">20" in condition S$_2$ are not inclusive. So they can be triggered at the same time so that the value of the common field is actual; the temperature in actionS$_1$ is equal to the temperature in action S$_2$, but the heater in actionR1 is not similar to the air_conditioner in condition S$_2$ and the ">27" in actionS$_1$ is the same as "< 15" has no intersection, so the value of the MutVal field is valid. After the rule conflict detection, the two rules conform to the formal expression of negative impact conflict, and the output rules S$_1$and S$_2$ have negative impact conflict. Since rule conflicts can be detected without analyzing rule S$_5$, S$_5$ is no longer described. Through the rule interaction analysis, Light [Light], which is on in conditionS$_3$ is equal to Light [Light], is on in conditionS$_4$, but "=1" in conditionS$_3$ and "=0" in conditionR4 are not inclusive, so they can trigger at the same time so that the value of the Com field is actual; the curtain, is in actionS$_3$ is equal to the curtain, is in actionS$_4$, so the area Sam Share Sub is valid; "=0" in actionS$_3$ is not equal to "=1" in conditionS$_4$ " makes the field DifAct evaluate to true. After the rule conflict detection, the two rules conform to the formal expression of execution conflict, and the output rules S$_3$ and S$_4$ have execution conflict. The above rule detection process does not depend on the natural environment, so that that rule conflicts can be detected in any environment, however, the three methods compared in this paper need to see the clashes between these two rules under certain conditions.

4. Experiment and result analysis

4.1. Research questions

To verify the effectiveness of the proposed method, three research questions are posed in this paper as follows:

RQ1: Compared with the existing conflict detection methods, whether the method in this paper can detect more comprehensive types of rule conflicts.
RQ2: Compared with the existing conflict detection methods, whether the detection results of this method are better.
RQ3: Conduct a comparative experiment of the method itself, and verify the effectiveness of the method by deleting a certain part or several parts of the method.

4.2. Subjects

The experimental objects of this paper are two IIOT systems, namely, the intelligent conference room simulation system and the intelligent fishery simulation system. The introduction of these two systems and the rules contained in them are shown in Table 3 and

Table 3
Introduction of Intelligent Meeting Room System

Intelligent Meeting Room System	Rule Type	Number Of Lines	Rule Description	Design Principles
Using the Internet of Things technology, a simulation system that controls the conference room environment and multimedia equipment through rules.	Execute coverage	20	Each type of rule contains 6 relational operators a =, ≠, <, >, ≤, ≥. 2 logical operators and, or. Since illogicality can be expressed by the relational operator ≠, illogicality is not included here. Parameter values include boundary values and intermediate values.	1) According to the principles of equivalence class division and boundary value analysis in software testing technology, design representative rules. 2) Initialize different scenarios to ensure that the above there are types of rules that are triggered to execute.
	Implementation contradictions	22		
	Negative impacts	24		
	Exclusive resources	28		
	Directly ignore dependencies	26		
	Direct circular dependency	32		
	Indirect circular dependency	45		
	Scene initialization	91		

Table 4.

4.3. Experimental environment and parameter settings

The software environment of the experiment is Jdk 1.8, Maven 3.6, IntelliJIDEA 2020.1, and Drools 7.4. In addition, the experimental data is processed in this paper, and 80% of the rule files are randomly selected from each rule base for the three experiments to avoid the uneven distribution of rule conflict types.

4.4. Evaluation indicators

Before calculating the evaluation index, it is necessary to calculate the confusion matrix of the multiclassification problem. Each element $d_{ij}(i, j \in N)$ of the confusion matrix represents the actual category i of the sample, and the classifier determines the count of category j. Then calculate the dichotomous confusion matrix elements B, c, d,e that belong to the sample x_i of type I, and finally calculate the evaluation index $G1(x_i)$, where

$$b = \sum_{i=1}^{m} d_{ii}, c = \sum_{i=1}^{m} \sum_{l=1,l\neq i}^{m} d_{il} d \sum_{i=1}^{m} \sum_{k=1,k\neq i}^{m} d_{ki}$$

$$e = \sum_{k=1,k\neq i}^{m} \sum_{l=1,l\neq i}^{m} d_{kl} d(i,k,l \in \mathbb{N})$$

$$S(x_i) = b/(b+c) \tag{1}$$

$$Q(x_i) = b/(b+d), \tag{2}$$

$$G1(x_i) = \frac{2 \times Q(x_i) \times S(x_i)}{Q(x_i) + S(x_i)} \tag{3}$$

In the experiment, the rule conflict detection method is implemented. Then, the rule conflict results obtained by the algorithm and the actual rule conflict results are quantified using the above evaluation indicators.

4.5. Experimental results and analysis

1) RQ1: Compared with the existing conflict detection methods, whether the method in this paper can detect more comprehensive types of rule conflicts. To explore whether the method in this paper can detect the types of rule conflicts more comprehensively, the FRCD method is compared with the three existing rule conflict detection methods. The experimental results are shown in Table 5. FRCD detected 7 kinds of conflicts, UTEA detected 3 types of conflicts, SPIDER detected four, and IRIS caught 3 kinds of conflicts.

The SPIDER supports direct circular dependency, but it was not detected in this experiment. Because the SPIDER algorithm detects the conflict when the system enters the conflict state, but the Internet of Things system in this experiment executes these two types of rules. As a result, it will join a deadlock state, causing the system to crash. So, this conflict detection method is not applicable. This paper investigates and analyzes the existing rules of the Internet of Things system and summarizes direct and indirect circular dependency conflicts according to the loop structure between the governments. Redundancy between authorities sums up the execution coverage conflicts.

Table 4
Introduction of the Intelligent Fishery System

Intelligent Fishery System	Rule Type	Number Of Lines	Rule Description	Design Principles
Using the Internet of Things technology, a simulation system that controls ship navigation and fish fishing through rules.	Execute coverage	22	Each type of rule contains 6 relational operators a =, ≠, <, >, ≤, ≥. 2 logical operators and, or. Since illogicality can be expressed by the relational operator ≠, illogicality is not included here. Parameter values include boundary values and intermediate values.	1) According to the principles of equivalence class division and boundary value analysis in software testing technology, design representative rules. 2) Initialize different scenarios to ensure that all the above types of rules are triggered and executed.
	Implementation Contradictions	26		
	negative impacts	20		
	Exclusive resources	24		
	directly ignore dependencies	28		
	direct circular dependency	30		
	indirect circular dependency	33		
	scene initialization	86		

Table 5
Whether conflict types can be detected.

Conflict Type	FRCD	UTEA	SPIDER	IRIS
Execution Coverage	√	√	√	√
Implementation Contradictions	√	√	√	√
Negative Impact	√	√	√	√
Exclusive Resources	√	x	x	x
Directly Ignore Dependencies	√	x	x	x
Direct Circular Dependency	√	△	△	x
Indirect Circular Dependency	√	△	x	x

1) RQ2: Compared with the existing conflict detection methods, whether the G1 value of the detection results of the way in this paper is higher. To explore whether the G1 value of the detection results of the method in this paper is higher, the method FRCD in this paper is compared with the three existing rule conflict detection algorithms.

The experimental comparison results of the simulated intelligent conference room system are shown in Table 6 and Fig. 6. The FRCD algorithm detects seven conflicts, covering all conflict types, and their G1 index average value is 78.3%. UTEA sees three kinds of conflicts, execution coverage, execution contradiction, and negative impact, and their G1 index average value is 68.6%. SPIDER catches four types of disputes, execution, coverage, execution, contradiction, adverse effect, and direct dependency neglect, and their G1 index average value is 57.9%. Finally, IRIS detects three kinds of conflicts, execution coverage, execution contradiction, and negative impact, and their F1 index average value is 69.0%. In the intelligent conference room system, the detection effect of the FRCD algorithm exceeds the other three algorithms. The experimental comparison results of the simulated intelligent fishery system are shown in Table 7 and Fig. 7. The FRCD algorithm detects seven conflicts, covering all conflict types, and their G1 index average value is 90.9%. UTEA sees three kinds of conflicts, execution coverage, execution contradiction, and negative impact, and their G1 index average value is 87.1%. SPIDER detects four types of disputes, execution coverage, execution contradiction, adverse effect, and direct dependency neglect, and their G1 index average value is 63.1%. Finally, IRIS detects three kinds of conflicts, execution coverage, execution contradiction, and negative impact, and their G1 index average value is 77.5%. On the intelligent fishery ship system, the detection effect of the FRCD algorithm exceeds that of the other three algorithms.

This paper further analyzes why the detection effect of FRCD is better than the other three methods. Among them, UTEA needs the scenario of initialization rules, but FRCD does not, so it can avoid making insufficient initialization scenarios and affecting the conflict detection effect. In addition, two kinds of conflict, static detection, direct circular dependency and indirect circular dependency, are implemented to prevent the system from entering a deadlock state due to circular dependency rules. In addition, the FRCD algorithm considers continuous system variables, but SPIDER and IRIS only consider discrete system states in rules. However, in the IIOT system corresponding to the experimental case in this paper, there are continuous system states, so it is necessary to detect these steady system states. The FRCD approach is contrasted with the three other rule conflict detection methods to see if the method in this paper can more thoroughly identify the different sorts of rule conflicts. Seven conflicts including all conflict kinds are found by the FRCD method, and their average G1 index value is 90.9%. It can be inferred that the approach in this study can detect more comprehensive sorts of rule conflicts as the FRCD method has carried out a full summary of conflict types for each element of the rules of the Internet of Things system. The FRCD algorithm's detection performance is superior to that of the other three algorithms on the intelligent fisheries ship system. Through the above experimental analysis, it can be concluded that the G1 value of the detection results of the method in this paper is higher.

1) RQ3: Verify the method's effectiveness by gradually deleting a specific part or several parts of the technique.

The method's effectiveness is verified by gradually deleting a specific part or several parts of the technique. In this paper, on the simulated intelligent conference room system, the experiment is carried out by deleting the rule interaction relations defined in Table 1 one by one. Among them, $RC(S_i, S_j)$ and $OR(S_i, S_j)$ do not participate in the deletion because removing any one of them will cause a specific type of conflict to be undetectable. The experimental results are shown in Table 8 and Fig. 8; FRCD is the highest of all other conflict indicators.

FRCD-CC means that the CC part is removed, except for the evaluation indicators of execution coverage, execution conflict, and only resource type; the other conflict types are reduced. Because CC is used to infer whether two rules are triggered in the same scene, if it is false, it will not enter into the matching of its related conflict type. If it evaluates to true, the conflict type related to it will be further detected. Removing the CC part means that the value of CC is always authentic, leading to errors in detecting specific rules. FRCD-SSS means to remove the SSS part, and FRCD-SMS means to remove the SMS part, and the detection indexes of these two experiments are all reduced. In addition, because these two experiments did not consider whether the device can be shared, some rule conflicts involving device concurrency cannot be detected, resulting in a decrease in the conflict detection index. FRCD-SA means to remove the SA part, and FRCD-DA means to remove the DA part. As a result, the detection indexes of these two experiments, except for the evaluation index of the negative impact type, are all reduced. Furthermore, because these two experiments did not consider all action types of the rules, SA and DA were considered at the same time to form the complete set of rule actions, they only detected a part

Table 6
Comparison of accuracy, recall, and G1 value results of different methods of intelligent conference room system

Conflict type	FRCD			UTEA			SPIDER			IRIS		
	Acc	Recall	G1	Acc	Recall	G1	Acc	Recall	G1	Acc	Recall	G1
Execute coverage	81.8	90	85.7	100	80	84.2	37	90	52.6	45.5	90.9	60
Implementation contradictions	91.7	100	95.7	83.3	90.9	58.8	72.7	42.1	53.3	83.3	47.6	60
Negative impacts	100	71.4	83.3	100	91.7	62.9	100	83.3	90.9	100	76.9	87
Exclusive resources	100	85.7	92.3	x	x	x	x	x	x	x	x	x
Directly ignore dependencies	35.5	91.7	51.2	x	x	x	50	26.7	34.8	x	x	x
Direct circular dependency	100	47.8	64.7	△	△	△	△	△	△	x	x	x
Indirect circular dependency	100	60	75	△	△	△	x	x	x	x	x	x

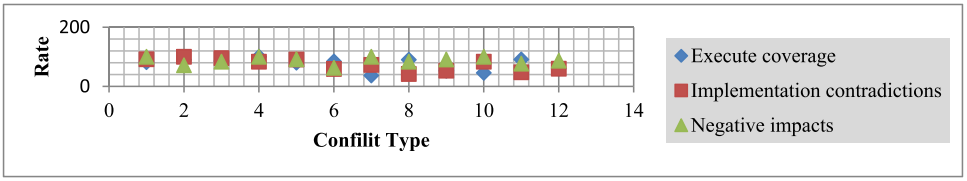


Fig. 6. Comparison of accuracy, recall, and G1 value results of different methods of intelligent conference room system

Table 7
Comparison of accuracy, recall, and G1 value results of different methods for intelligent fishing vessel system

Conflict type	FRCD			UTEA			SPIDER			IRIS		
	Acc	Recall	G1	Acc	Recall	G1	Acc	Recall	G1	Acc	Recall	G1
Execute coverage	100	100	100	100	81.8	90	45.5	90.9	60.6	43.5	90.9	83.3
Implementation contradictions	100	100	100	85.7	92.3	88.9	100	48	64.9	100	50	66.7
Negative impacts	88.9	80	84.2	100	70	82.4	100	70	82.4	100	70	82.4
Exclusive resources	100	83.3	90	x	x	x	x	x	x	x	x	x
Directly ignore dependencies	88.9	85.7	87.3	x	x	x	100	28.6	44.4	x	x	x
Direct circular dependency	100	80	88.9	△	△	△	△	△	△	x	x	x
Indirect circular dependency	100	75	85.7	△	△	△	x	x	x	x	x	x

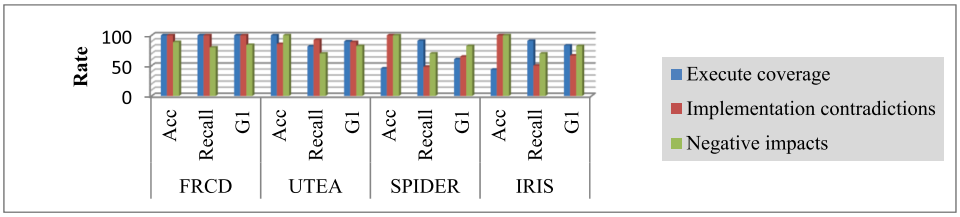


Fig. 7. Comparison of accuracy, recall, and G1 value results of different methods for intelligent fishing vessel system

Table 8
Comparison of the F1 value results of different methods for reducing the interaction relationship rules

Conflict type	FRCD	FRCD- CC	FRCD- SSS	FRCD- SMS	FRCD- SA	FRCD- DA	FRCD- MV
Execute coverage	85.7	85.7	58.1	60	45.5	60	60
Implementation Contradictions	95.7	95.7	61.1	64.7	64.7	48.9	64.7
Negative impacts	83.3	80	60.6	58.8	80	83.3	31.2
Exclusive resources	92.3	92.3	63.2	40	25	83.3	63.2
Directly ignore dependencies	51.2	48.9	48.9	46.8	40.7	39.3	38.6
Direct circular dependency	64.7	57.9	57.1	57.1	57.1	59.5	57.9
Indirect circular dependency	75	70.6	68.6	68.6	68.6	70.6	66.7

of the action types, resulting in a decrease in the conflict detection index.

FRCD-MV means removing the MV part, and its detection indicators are reduced because this experiment did not consider the conflict caused by different rules affecting the same attribute value, resulting in a decrease in the conflict detection index. After

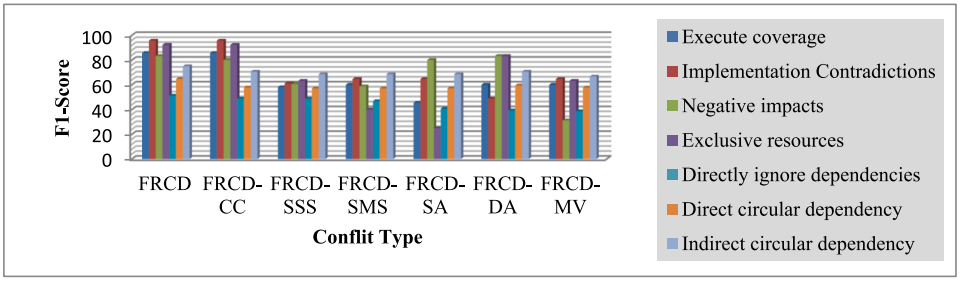


Fig. 8. Comparison of the F1 value results of different methods for reducing the interaction relationship rules

analyzing the above experimental results, it can be concluded that the method of this paper must consider all rule interaction parts.

5. Conclusion

The rules in the control logic for the core component system of Internet of Things system design are proposed in this study as a conflict detection approach. This method employs a formal process to model the rules and various rule conflicts in the Internet of Things after first analyzing and categorizing the rules of the system. For various rule conflicts, their traditional expressions are other, so that different rule conflicts can be detected clearly. Then, this method can analyze the input Internet of Things rules, get various conditions and actions of the authorities, and decompose these conditions based on the analysis results, which can help simplify the logic of the rules and requirements. Finally, the corresponding conflicts are detected according to different rule conflict types. The experiments carried out in two Internet of Things systems in this paper are compared with three existing Internet of Things rule conflict detection methods. The experimental findings demonstrate that this method has the best rule conflict detection effect compared to the others. The approach outlined in this study could potentially be improved. The IIOT rule conflict type analysis, for instance, is not thorough enough. The algorithm, however, is only able to indicate which rules are in conflict; it is unable to offer advice or suggestions for improving the correct rules. Additionally, there is no visual representation of how many regulations interact in conflict. All these things need to be researched for upcoming work.

Funding statement

This work does not receive any kind of funding in any form.

Declaration of Competing Interest

The authors declare no conflict of interest.

Data availability

Data will be made available on request.

References

[1] Lv B, Sun R, Zhang H, Xu H, Yue R. Automatic vehicle-pedestrian conflict identification with trajectories of road users extracted from roadside LiDAR sensors using a rule-based method. IEEE Access 2019;7:161594–606. https://doi.org/10.1109/ACCESS.2019.2951763.
[2] Ramasso E, Rombaut M, Pellerin D. State filtering and change detection using TBM conflict application to human action recognition in athletics videos. IEEE Trans Circuits Syst Video Technol 2007;17(7):944–9. https://doi.org/10.1109/TCSVT.2007.896652. July.
[3] Maherin I, Liang Q. Multistep Information Fusion for Target Detection Using UWB Radar Sensor Network. IEEE Sensors J 2015;15(10):5927–37. https://doi.org/10.1109/JSEN.2015.2451160. Oct.
[4] Xiong J, Zhang Q, Peng Z, Sun G, Cai Y. Double Sample Data Fusion Method Based on Combination Rules. IEEE Access 2016;4:7487–99. https://doi.org/10.1109/ACCESS.2016.2604824.
[5] Wu Chih-Hung, Lee Shie-Jue. Enhanced high-level Petri nets with multiple colors for knowledge verification/validation of rule-based expert systems. In: IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). 27; Oct. 1997. p. 760–73. https://doi.org/10.1109/3477.623230.
[6] Gopalakrishnan T, Ruby D, Al-Turjman F, Gupta D, Pustokhina IV, Pustokhin DA, Shankar K. Deep learning enabled data offloading with cyber attack detection model in mobile edge computing systems. In: Institute of Electrical and Electronics Engineers (IEEE). 8. IEEE Access; 2020. p. 185938–49. https://doi.org/10.1109/access.2020.3030726.
[7] Cuadrado JS, Guerra E, de Lara J. Static Analysis of Model Transformations. IEEE Trans Software Eng 2017;43(9):868–97. https://doi.org/10.1109/TSE.2016.2635137. 1 Sept.
[8] Tharewal Sumegh, Ashfaque Mohammed Waseem, Banu Sayyada Sara, Uma Perumal, Hassen Samar Mansour, Shabaz Mohammad. Intrusion Detection System for Industrial Internet of Things Based on Deep Reinforcement Learning, 2022. Wireless Communications and Mobile Computing; 2022. p. 8. https://doi.org/10.1155/2022/9023719. Article ID 9023719pages.
[9] Ma S, Yuan Y, Wu J, Jiang Y, Jia B, Li W. Multisensor decision approach for HVCB fault detection based on the vibration information. IEEE Sensors J 2021;21(2):985–94. https://doi.org/10.1109/JSEN.2020.2980081. 15 Jan.15.

[10] Choi Young Sik, Krishnapuram R. A robust approach to image enhancement based on fuzzy logic. IEEE Trans Image Process 1997;6(6):808–25. https://doi.org/10.1109/83.585232. June.
[11] Borst C, Visser RM, van Paassen MM, Mulder M. Exploring short-term training effects of ecological interfaces: a case study in air traffic control. IEEE Trans Human-Mach Syst 2019;49(6):623–32. https://doi.org/10.1109/THMS.2019.2919742. Dec.
[12] Alqaralleh BAY, Mohanty SN, Gupta D, Khanna A, Shankar K, Vaiyapuri T. ReliaBLE MULTI-OBJECT TRACKING MODEL USING DEEP LEARNING AND ENERGY EFFICIENT WIRELESS MULTIMEDIA SENSOR NETWorks. IEEE Access, 8. Institute of Electrical and Electronics Engineers (IEEE); 2020. p. 213426–36. https://doi.org/10.1109/access.2020.3039695.
[13] Jiayun C, Xiongjun F, Wen J, Min X. Design of high-performance energy integrator detector for wideband radar. J Syst Eng Electron 2019;30(6):1110–8. https://doi.org/10.21629/JSEE.2019.06.07. Dec.
[14] Gao F, Li Y, Lu S. Extracting Moving Objects More Accurately: A CDA Contour Optimizer. IEEE Trans Circuits Syst Video Technol 2021;31(12):4840–9. https://doi.org/10.1109/TCSVT.2021.3055539. Dec.
[15] Zhao S, Luo Y, Zhang T, Guo W, Zhang Z. A Feature Decomposition-Based Method for Automatic Ship Detection Crossing Different Satellite SAR Images. IEEE Trans Geosci Remote Sens 2022;60:1–15. https://doi.org/10.1109/TGRS.2022.3201628. Art no. 5234015.
[16] Matalas L, Benjamin R, Kitney R. An edge detection technique using the facet model and parameterized relaxation labeling. IEEE Trans Pattern Anal Mach Intell 1997;19(4):328–41. https://doi.org/10.1109/34.588006. April.
[17] Deshmukh Shyam, Rao Komati Thirupathi, Shabaz Mohammad. Collaborative Learning Based Straggler Prevention in Large-Scale Distributed Computing Framework, 2021. Security and Communication Networks; 2021. p. 9. https://doi.org/10.1155/2021/8340925. Article ID 8340925pages.
[18] Zhang N, Liu Y, Xu L, Lin P, Zhao H, Chang M. Magnetic Anomaly Detection Method Based on Feature Fusion and Isolation Forest Algorithm. IEEE Access 2022; 10:84444–57. https://doi.org/10.1109/ACCESS.2022.3197630.
[19] Kokkinos E, Maras AM. Locally optimum Bayes detection in nonadditive first-order Markov noise. IEEE Trans Commun 1999;47(3):387–96. https://doi.org/10.1109/26.752819. March.
[20] Borkar A, Donode A, Kumari A. A survey on intrusion detection system (IDS) and internal intrusion detection and protection system (IIDPS). In: 2017 International Conference on Inventive Computing and Informatics (ICICI); 2017. p. 949–53. Nov.
[21] Jiang Y, Wang W, Zhao C. A machine vision-based realtime anomaly detection method for industrial products using deep learning. In: 2019 Chinese Automation Congress (CAC); 2019. p. 4842–7. Nov.
[22] Istiake Sunny MA, Maswood MMS, Alharbi AG. Deep learning-based stock price prediction using LSTM and bi-directional LSTM model. In: 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES); 2020. p. 87–92. Oct.
[23] Chen L, Kuang X, Xu A, Suo S, Yang Y. A novel network intrusion detection system based on CNN. In: 2020 Eighth International Conference on Advanced Cloud and Big Data (CBD); 2020. p. 243–7. Dec.
[24] Patil YS, Sankpal SV. EGSP: enhanced grid sensor placement algorithm for energy theft detection in smart grids. In: 2019 IEEE 5th International Conference for Convergence in Technology (I2CT); 2019. p. 1–5. March.
[25] Yuan X, Chen J, Zhang N, Fang X, Liu D. A federated bidirectional connection broad learning scheme for secure data sharing in Internet of Vehicles. China Commun 2021;18(7):117–33.
[26] Suwannalai E, Polprasert C. Network intrusion detection systems using adversarial reinforcement learning with deep Qnetwork. In: 2020 18th International Conference on ICT and Knowledge Engineering (ICT&KE); 2020. p. 1–7. Nov.
[27] Zhou W, Li J, Chen Y, Shen L. Strategic interaction multi-agent deep reinforcement learning. IEEE Access 2020;8:119000–9.
[28] Wu B, Feng Y. Policy reuse for learning and planning in partially observable Markov decision processes. In: 2017 4th International Conference on Information Science and Control Engineering (ICISCE); 2017. p. 549–52. July.
[29] Wang D, Ding B, Feng D. Meta reinforcement learning with generative adversarial reward from expert knowledge. In: 2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education (ICISCAE); 2020. p. 1–7. Sept.

Pradyumna Kumar Tripathy has completed his M.Tech. and Ph.D. in Computer Science from Utkal University, India. He is currently working as Associate Professor in the Department of Computer Science & Engineering at Silicon Institute of Technology, Bhubaneswar, India. His research interests include Reliability Analysis of Interconnection Networks, Parallel Distributed Systems, Topological Optimization of Interconnection Networks, IoT and Machine Learning.

Mohammad Shabaz is working as Assistant Professor at the Model Institute of Engineering and Technology, Jammu, J&K, India. He specializes in Multimedia-based Emerging Technologies and Data Analytics for Neuroscience as a Service (NaaS), Complex Computational Analysis, and Algorithm designing. He is an editor of Neuroscience Informatics. He has published more than 200 scientific peer-reviewed articles indexed in international databases.

Abdelhamid ZAIDI is an Assistant Professor in the College of Science at Qassim University in Saudi Arabia. He has a PhD in Statistics from University Grenoble-Alpes (France) and an Engineering degree in Computer Science and Applied Mathematics from ENSIMAG Grenoble (France). He works mainly on the development of computational methods applied to various subjects of signal and image processing.

Ismail Keshta received his B.Sc. and the M.Sc. degrees in computer engineering and his Ph.D. in computer science and engineering from the King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia. He is currently an Assistant Professor in AlMaarefa University, Riyadh, Saudi Arabia. His research interests include software process improvement, modeling, and intelligent systems.

Uttam Sharma (IT Analyst, TCS) received a Ph.D. degree in Computer Science and Engineering from the Gautam Buddha University, India in 2023 He is currently an I.T. Analyst at TCS (Tata Consultancy Services). His current research interests are in the areas of Fingerprint Template Security, 3D Ear Biometric, Pattern Recognition, Artificial Intelligence and Deep Learning, and the Internet of Things.

Mukesh Soni has completed his Bachelor's in Information Technology from Gyan Ganga Institute of Technology & Management, Bhopal, India and Masters in Computer Science & Engineering from MANIT, Bhopal, India. He is currently Research Scholar at Department of Computer Science & Engineering, Jagran Lake city University, Bhopal. His research interests include Applied Cryptography, Information Security, and Network Security.

Anurag Vijay Agrawal (Senior Member, IEEE) received the Ph.D. degree in Electronics and Communication Engineering from the Indian Institute of Technology Roorkee, India in 2021. His current research interests are in the areas of MIMO/Massive MIMO communications, digital predistortion, energy efficiency, transportation engineering, high-speed Railways communications, Intelligent Transportation Systems, 5G/6G Signal Generation & Enhancement and Internet of Things.

Renato Racelis Maaliw III is an Associate Professor in College of Engineering in Southern Luzon State University, Lucban, Quezon, Philippines. He has a doctorate degree in Information Technology and Master's degree in Information Technology with specialization in Web Technologies. His area of interest is in computer engineering, web technologies, software engineering, data mining, machine learning, and analytics.

D P Sharma working as Director, MUJ-TEC & Professor in the department of IT, School of Computing and Information Technology at Manipal University Jaipur, India. He has Experience of 21+ Years. He has Authored 1 Text Books, 10 Articles.

Computers and Electrical Engineering • Volume 108 • May 2023 • Article number 108671

Document type
Article

Source type
Journal

ISSN
00457906

DOI
10.1016/j.compeleceng.2023.108671

Policy conflict detection approach for decision-making in intelligent industrial Internet of Things

Tripathy, Pradyumna Kumar[a]; Shabaz, Mohammad[b] ✉; Zaidi, Abdelhamid[c]; Keshta, Ismail[d]; Sharma, Uttam[e]; Soni, Mukesh[f]; Agrawal, Anurag Vijay[g]; Maaliw III, Renato R.[h]; Sharma D.P.[i]

[a] Department of Computer Science and Engineering, Silicon Institute of Technology, Bhubaneswar, Odisha

[b] Model Institute of Engineering and Technology, J&K, Jammu, India

[c] Department of mathematics, College of Science, Qassim University, P.O.Box 6644, Buraydah, 51452, Saudi Arabia

[d] Computer Science and Information Systems Department, College of Applied Sciences, AlMaarefa University, Riyadh, Saudi Arabia

Abstract

Improving Industrial Internet of Things (IIOT), device flexibility and lowering maintenance costs are significant problems. However, as the scale of the Intelligent Industrial Internet of Things (IIIOTs) system expands, the interactions between the rules get more sophisticated, potentially leading to rule discrepancies. The algorithm for Formal Rule Conflict Detection (FRCD) is created, and a thorough explanation of the procedure is given in this paper. Two IIIOT systems were used in experiments and the results were compared with three existing standard IIIOT rule conflict detection techniques. They include policy conflict detection systems based on Web semantics (Semantic Web-Based Policy Interaction Detection with Rules, (SPIDER)), conflict detection methods based on Users, Triggers, Environment entities, and Actuators (UTEA), and semiformal conflict detection methods (Identifying Requirements Interactions using Semiformal (IRIS)). The experimental results show that the FRCD rule conflict detection method is superior. © 2023 Elsevier Ltd

Author keywords

Artificial intelligent; Decision making; Intelligent industrial Internet of Things; Internet of Things; Policy conflict; Rule conflict detection

ELSEVIER
Computers and Electrical Engineering
Volume 108, May 2023, 108671

Policy conflict detection approach for decision-making in intelligent industrial Internet of Things

Pradyumna Kumar Tripathy [a], Mohammad Shabaz [b], Abdelhamid Zaidi [c], Ismail Keshta [d], Uttam Sharma [e], Mukesh Soni [f], Anurag Vijay Agrawal [g], Renato R. Maaliw III [h], D.P. Sharma [i]

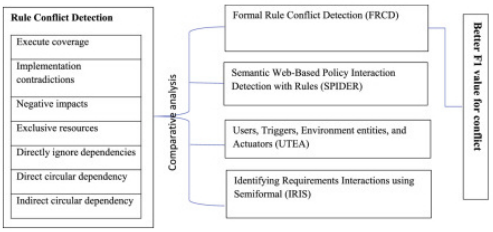
Show more

https://doi.org/10.1016/j.compeleceng.2023.108671

Abstract

Improving Industrial Internet of Things (IIOT), device flexibility and lowering maintenance costs are significant problems. However, as the scale of the Intelligent Industrial Internet of Things (IIIOTs) system expands, the interactions between the rules get more sophisticated, potentially leading to rule discrepancies. The algorithm for Formal Rule Conflict Detection (FRCD) is created, and a thorough explanation of the procedure is given in this paper. Two IIIOT systems were used in experiments and the results were compared with three existing standard IIIOT rule conflict detection techniques. They include policy conflict detection systems based on Web semantics (Semantic Web-Based Policy Interaction Detection with Rules, (SPIDER)), conflict detection methods based on Users, Triggers, Environment entities, and Actuators (UTEA), and semiformal conflict detection methods (Identifying Requirements Interactions using Semiformal (IRIS)). The experimental results show that the FRCD rule conflict detection method is superior.

Graphical abstract



Introduction

The Internet of Things refers to the Internet between objects and objects. Through wireless sensing technology, it uses sensors to obtain information about objects and the environment. It realizes information transmission and resource sharing between physical devices and between physical devices and networks [1]. Fig. 1 is a typical IIOT system architecture, mainly divided into three parts: external elements, network layer, and control system. Exterior features include sensors and devices. Sensors are the source of information flow and can collect physical quantities such as temperature, humidity, light intensity, and pressure. Devices include programmable hardware and are destinations for information flow. IoT sensors and actuators are part of an IoT subsystem, which is controlled by one or more

Reviewer 1:

Comments:

1. The paper presents an interesting and timely topic in the field of intelligent industrial IoT. The policy conflict detection approach discussed is relevant and important for decision-making processes.
2. The introduction provides a clear background and motivation for the research, establishing the significance of the problem.
3. However, the methodology section needs more details on the specific algorithms or techniques used for policy conflict detection. Please provide more information to enhance the reproducibility of the study.
4. The results and discussion sections are well-organized and provide a comprehensive analysis of the proposed approach. However, it would be helpful to include some real-world case studies or experiments to validate the effectiveness of the approach.
5. The conclusion is concise and summarizes the key findings. Consider adding future research directions to further strengthen the paper.

Reviewer 2:

Comments:

1. The paper addresses an important area of research in the intelligent industrial IoT domain. The topic of policy conflict detection is relevant and can contribute to effective decision-making processes.
2. The methodology section is well-presented and provides a clear understanding of the approach used. The use of a rule-based system and machine learning techniques is appropriate for this study.
3. The experimental setup and evaluation are well-described, and the results demonstrate the effectiveness of the proposed approach. The comparison with existing methods adds value to the paper.
4. However, the discussion lacks a deeper analysis of the limitations and potential challenges of the proposed approach. Consider expanding this section to provide a more comprehensive view.
5. Overall, this paper is a valuable contribution to the field, and with some minor revisions, it can be suitable for publication.

Reviewer 3:

Comments:

1. The paper explores an interesting area in the intelligent industrial IoT domain. The topic of policy conflict detection is relevant and has practical implications for decision-making.
2. The introduction provides a clear background and motivation for the study. The problem statement is well-defined, highlighting the significance of addressing policy conflicts in decision-making processes.
3. The methodology section lacks clarity on the implementation details of the proposed approach. Please provide more information on the specific steps and algorithms used for policy conflict detection.
4. The results and discussion sections present a comprehensive analysis of the proposed approach. However, it would be beneficial to include some visual representations, such as graphs or tables, to enhance the clarity of the results.

FACULTY POSITION RECLASSIFICATION FOR SUCS

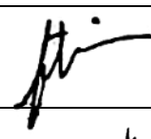
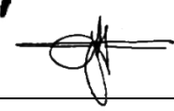
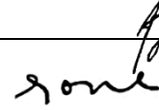
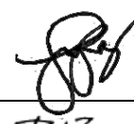
(DBM-CHED Joint Circular No. 3, series of 2022)

CERTIFICATION OF PERCENTAGE CONTRIBUTION

(Research Output with Multiple Authors)

Title of Research: Policy Conflict Detection Approach over Intelligent Industrial Internet of Things for Decision-Making
Type of Research Output: Journal Paper (Scopus-Indexed, Elsevier Paper)

Instruction: Supply ALL the names of the authors involved in the publication of Research output and indicate the contribution of each author in percentage. Each author shall sign the conforme column if he/she agrees with the distribution. The conforme should be signed by all the authors in order to be considered. Please prepare separate Certification for each output.

	Name of Authors	% Contribution	Conforme *(Sign if you agree with the % distribution)*
1	Pradyumna Kumar Tripathy	20%	
2	Mohammad Shabaz	10%	
3	Abdelhamid Zaidi	10%	
4	Ismail Keshta	10%	
5	Uttam Sharma	10%	
6	Mukesh Soni	10%	
7	Anurag Vijay Agrawal	10%	
8	Renato R. Maaliw III	10%	
9	D. P. Sharma	10%	
	** Should have a total of 100%*	**100.00%**	

Prepared by:

Renato R. Maaliw III, DIT
(Name and Signature)
Faculty

Certified by:

Nicanor L. Guinto, Ph.D
(Name and Signature)
Director, Office of Research Services