Computing and Informatics, Vol. *, 2022,**

MODELING AND ANALYZING USER BEHAVIOR RISKS IN ONLINE SHOPPING PROCESSES BASED ON DATA-DRIVEN AND PETRI-NET METHODS

Wangyang Yu[1,2], Zhuojing Ma[2], Xiaojun Zhai[3,*],
Yuke Zhou[4,*], Weiwei Zhou[5], and Yuan Liu[2]

[1] *The Key Laboratory of Modern Teaching Technology, Ministry of Education, Xi'an, China*
[2] *School of Computer Science, Shaanxi Normal University, Xi'an, China*
[3] *School of Computer Science and Electronic Engineering, University of Essex, UK*
[4] *Key Laboratory of Ecosystem Network Observation and Modeling, Chinese Academy of Sciences, China*
[5] *Business School, Shandong Yingcai University, China*

e-mail: `xzhai@essex.ac.uk`

Manuscript received ...
Communicated by...

Abstract. With the rapid spread of e-commerce and e-payment, the increasing number of people choose online shopping instead of traditional buying way. However, the malicious user behaviors have a significant influence on the security of users' accounts and property. In order to guarantee the security of shopping environment, a method based on Complex Event Process (CEP) and Colored Petri nets (CPN) is proposed in this paper. CEP is a data-driven technology that can correlate and process a large amount of data according to Event Patterns, and CPN is a formal model that can simulate and verify the specifications of the online shopping processes. In this work, we first define the modeling scheme to depict the user behaviors and Event Patterns of online shopping processes based on CPN. The Event Patterns can be constructed and verified by formal methods, which guarantees the correctness of Event Patterns. After that, the Event Patterns are translated into Event Pattern Language (EPL) according to the corresponding algorithms. Finally, the EPLs can be inserted into the complex event processing engine to analyze the users' behavior flows in real-time. In this paper, we validate the effectiveness of the proposed method through case studies.

1 INTRODUCTION

In recent years, with the rapid development of Internet, online shopping has become a well-known way. As a novel mode, online shopping has a great impact on peoples' lifestyle and economic development. However, due to the virtuality, dynamic and open environment, the inherent defects of the software systems and network risks pose a great threat to the security of consumers' accounts and funds [1, 2, 3].

In order to improve the security of online shopping environment, some researches focus on authentication as the core security method, among which digital certificate [4], authentication technology [5] and dynamic verification code [6] are a most common method. Mining of user behavior data has been increasingly applied to the construction and analysis of user behavior patterns. Gull M. and Pervaiz A. builds user behavior patterns through data mining to analyze users' actual purchasing behaviors [7]. Some identity authentication methods are proposed by monitoring the behaviors of mobile devices [8]. As a distributed Web application, online shopping systems are loosely coupled and interactively complex. Despite the third party service providers bridging the gap of trustiness between merchants and users, their involvement complicates the logic flow in the checkout process [9]. Logic flows of online shopping systems allow malicious users to carry out malicious behaviors under legal identities, e.g., purchase products using fabricated payments [10, 11]. A user can abuse legitimate application-specific functionality against developers' intentions [12]. As a result, vulnerable servers are exposed to malicious users who can potentially implement the behaviors such as alternate the control and data flows through concurrent interactions [13].

With the development of data science, the use of machine learning to conduct real-time analysis of user behaviors has gradually gained more attention. Jiang et al. proposed the online detection methods for credit card fraud based on machine learning [14, 15, 16]. Credit card payment is an important component of the entire online shopping process. The online shopping process includes not only the payment, but also the place order, notification, confirmation, update information, and many other operations [17]. Guaranteeing the real-time security of the entire online shopping processes is the key of avoiding the frequent risks. Today's e-commerce businesses have become increasingly hybrid, with their program logic being distributed across multi-participants, including the servers and their clients, along with various third party API service providers [18]. Their respective business processes construct the entire transaction process. This integration introduces new security challenges due to complex interaction behaviors among multi-participants [19].

Therefore, the real-time identification of behavior risks in online shopping processes is imminent. In the process of risk prevention and control, we should fully consider the relationships among multiple events in the online shopping process, and dynamically identify users' risky behaviors in real-time. Real-time monitoring of the users' shopping data streams and intelligent identification of user behaviors can effectively improve the security of online shopping processes.

Complex Event Processing (CEP) is an an emerging reference framework and standard for building and managing event-driven information systems [20, 21]. The goal is to get the meaningful complex events by reasoning and analyzing the event data flow, and respond in real-time. The CEP framework includes Event Pattern construction and recognition, event association and abstraction, event-driven processing, etc.. CEP does not depend on specific methods and technologies, and many new theories, methods and technologies are needed to research and design specific CEP systems in a certain field. At present, it is widely used in the fields of business process analysis [22], financial analysis [23], RFID [24] and wireless sensor network [25].

In addition, the Event Patterns of most current researches on CEP are based on SQL-like statements and non-formal rules [26]. It is not enough to accurately describe the online shopping process which is distributed, complex, concurrent and loosely coupled. Specially, the correctness of Event Patterns and EPLs should be validated and guaranteed. Petri nets are a formal model that is suitable for portraying distributed systems that can accurately describe the concurrency and event relationships [27, 28], and widely used in Workflow [29], Web services [30], control systems [31,32]. Compared with non-formal rules, Petri nets have a wealth of analytical techniques and other derived advanced models (e.g., CPN), and can be applied to validate and analyze the Event Patterns formally and effectively, thus ensuring the correctness of the Event Patterns. CPN is a kind of high-level Petri nets with powerful graphical modeling ability for depicting discrete events systems. Meanwhile, it can effectively describe the complex structures in dynamic systems, such as sequence, concurrency, and selection. On the other hand, CPN has a mature visual modeling tool (CPN Tools). Ref. [33] has proposed a meaningful complex event processing model by Prioritized Colored Petri Net based on MEdit4CEP platform. Thus, CPN is an ideal model for depicting the users' risky behaviors of online shopping processes.

Therefore, in this paper, we coalesce the formal model (CPN) and data-driven framework (CEP) to construct the methodology for identifying user behavior risks of online shopping processes in real-time. The contributions of this paper mainly include:

- For accurately depicting and validating the Event Pattern of user behavior risks, this paper defines the formal modeling and validating scheme based on CPN.
- This paper proposes the algorithms for transforming the formal Event Pattern to EPL.
- This paper constructs the risk identification mechanism based on CPN and CEP to cope with the user behavior risks in online shopping processes.

The remainder of this paper is organized as follows. Section II introduces the related methods used in this paper. Section III illustrates the modeling principle and analyzing process of Event Patterns based on CPN. Section IV introduces the risk identification mechanism based on CPN and CEP, and the demo system of above methodology is implemented. Section VI concludes the paper.

2 RELATED METHODS

This section mainly introduces the related concepts and methods involved in the paper including CPN and CEP.

2.1 Colored Petri nets (CPN)

A CPN is a directed graph that combines the Petri net and the StandML (functional programming language). In CPN, a specific color set (data type) is provided for each place, the data types of the colored set mainly include int, boolean, string, list and record. We can also set the guard function and priority on the transition, and set the expressions on the arc in the process of constructing a CPN model. When the variables in the colored set satisfy the conditions of the input arc and the settings of the transition, the corresponding transition can be fired. More details on CPN can be seen in [27, 34].

Definition 1 [34] A Colored Petri Net is a nine-tuple $CPN = (P, T, A, \Sigma, V, C, G, E, I)$, where

1) P is a finite set of places.

2) T is a finite set of transitions such that $P \cap T = \emptyset$.

3) $A \subseteq P \times T \cup T \times P$ is a set of directed arcs.

4) Σ is a finite set of non-empty color sets.

5) V is a finite set of typed variables such that $Type[v] \in \Sigma$ for all variables $v \in V$.

6) $C : P \rightarrow \Sigma$ is a color set function that assigns a color set to each place.

7) $G : T \rightarrow EXPR_V$ is a guard function that assigns a guard to each transition t such that $Type[G(t)] = Bool$.

8) $E : A \rightarrow EXPR_V$ is an arc expression function that assigns an arc expression to each arc a such that $Type[E(a)] = C(p)_{MS}$, where p is the place connected to the arc a.

9) $I : P \rightarrow EXPR_\emptyset$ is an initialization function that assigns an initialization expression to each place p such that $Type[I(p)] = C(p)_{MS}$.

Definition 2 [34] A binding element $(t, b) \in BE$ is enabled in a marking M if and only if the following two properties are satisfied:

1) $G(t) \langle b \rangle$.

2) $\forall p \in P : E(p, t) \langle b \rangle \ll= M(p)$.

When (t, b) is enabled in M, it may occur, leading to the marking M' defined by:

3) $\forall p \in P : M'(p) = (M(p) - -E(p, t) \langle b \rangle) + +E(t, p) \langle b \rangle$.

2.2 Complex Event Process (CEP)

CEP is a real-time data processing framework, which is mainly used to research on how to efficiently extract valuable events from a large number of simple event streams, and can be abstracted and aggregated into complex events. It can quickly find the abnormal situation from the real-time data streams, which is suitable for the scene of abnormal detection [20, 23]. First, the acquired data flow (event stream) is captured by using filtering, association and aggregation; second, based on the temporal relation and aggregation relation among events, by developing the EPL, the valuable events (complex events) are continuously excavated from the event stream at different level, and then they can be abstracted and aggregated into high-level complex events; final, the highest-level complex events are responded by notifying the system, software, or device when a particular situation has been detected [26].

The so-called event means the meaningful state change in actual systems, usually divided into atomic events and complex events. Atomic events refer to the most basic information generated at a certain point-in-time in the process of system execution, which contains limited information and cannot be separated. Complex events means value ones those are generated by pattern matching of atomic events, as shown in Fig. 1. A complex event usually includes the time of occurrence, event attribute value and the event name. The Event Pattern is a template that is used to match the set of eligible event stream and accurately describes the causal, time and logical relationships among events. Event Pattern is mainly implemented by EPL, which is a SQL-like language with a rich set of advanced processing expressions. It provides a lot of times and pattern operators to define patterns of interest.

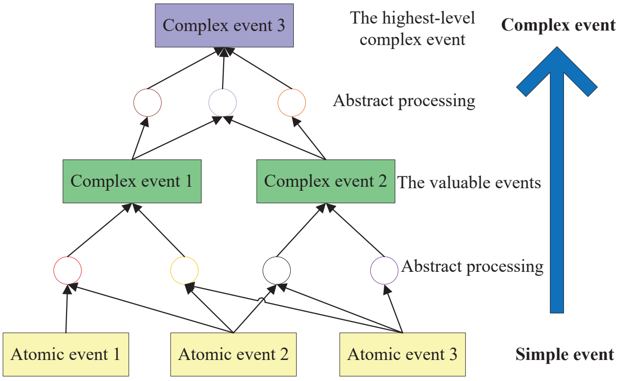


Fig. 1. The abstract process of complex events.

3 THE EVENT PATTERN MODELS OF BEHAVIOR RISK IDENTIFICATION

E-commerce business interaction is a typical distributed and concurrent system in the open Internet, the recent developments of which has opened a range of security challenges. A major reason is that the distributed system possesses concurrency and the execution may proceed in many different ways. A typical online distributed system handles process concurrency in a number of fashion. It is easy for a malicious user to implement behavior interactions during this process concurrency, which might lead to logical vulnerabilities in the system execution.

The identification models are the basis of analyzing the user behavior risks in online shopping processes. In this paper, a formal modeling method based on CPN is established to depict the risk behaviors of online shopping users, and the models can be validated to guarantee the correctness. In this section, the behavior risk identification models focus on the single-user and multi-user scenes. The risky behaviors includes but no limited to: the user's account address or payment method is abnormal in a short time; the abnormality of the payment amount is mainly reflected in two aspects: on the one hand, the user continuously places orders and the purchase amount continues to increase; on the other hand, the user's payment amount is greater than the average payment amount of the user over a period of time. The main colored sets and the variable declarations in Event Pattern models are shown in Table I.

Table 1. Main type definitions of the models

Colored sets	**Variable declarations**	**Implications**
colset Num=int	var n	The number of user behavior stream
colset Usern=string	var usern	The user name
colset Order=int	var order	The order number
colset Gross=int	var gross	The payment amount
colset Address=string	var place, address	User account address
colset Way=string	var way	The payment method
colset State=bool	var state	The payment result
colset Timee=int	var timee, timee1	The payment time

3.1 Modeling principles

In general, the operations of identifying user behavior risks mainly include input, filtering, aggregation, and analysis. However, the order of transition executions has great influence on the result. In order to reduce the adverse effects, it is necessary to set the transition priority in the process of modeling. According to the structure that can trigger transitions under a certain identification state and the region where it is located, the relevant principles are settled for the priority of transitions.

The meaning of the transition in CPN is usually a executable action, and the priority of the transition should meet the scenario setting of the online shopping process. Meanwhile, the transition structures in a certain marking of Event Pattern models are generally divided into the sequential, selection and concurrent structures, as shown in Fig. 2. The transition, which is marked red, indicates that it can be triggered in current state.

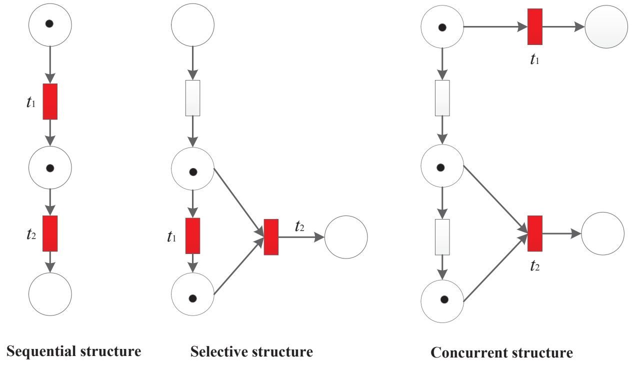


Fig. 2. The transition structures.

When we depict the model, if there is a sequence structure between two transitions that can be fired under a certain marking, setting different priorities for transitions will not affect the final result. If there is a selection structure between two transitions, the priority setting of the transitions has different effects on the final result. Therefore, in this scene, we should set the priority according to actual scene. If there is a concurrence structure between two transitions, the priority of the transitions do not affect the final result. The modeling scheme of Event Pattern models based on CPN is listed as follows:

- Online shopping user behaviors should be numbered in order, and ensure that transitions are triggered in order;
- The priority should be set according to the structure and region of the transition under a certain state;
- The setting of the guard function and arc expression need to satisfy the identification conditions of user behavior risks on the corresponding transition;
- When using the sliding or fixed window to analyze user behaviors, it's need to ensure the size of the window in real-time;
- When analyzing the user behavior risks, it is necessary to discharge the risk-free event streams (tokens) in real-time.

3.1.1 Case 1 - The Event Pattern model for the anomaly detection of the user's account address or the way of payment

This section illustrates the formal model of Event Pattern for abnormal detection of user account addresses, which mainly includes the filtering and identifying operations of online shopping behaviors. The model is shown in Fig. 3.

The filtering operation is mainly used to obtain the behavior streams whose statuses are paid in the entire online shopping behaviors. In the model, the behavior streams are represented by *colset TRAN1 = product INT * STRING * STRING **

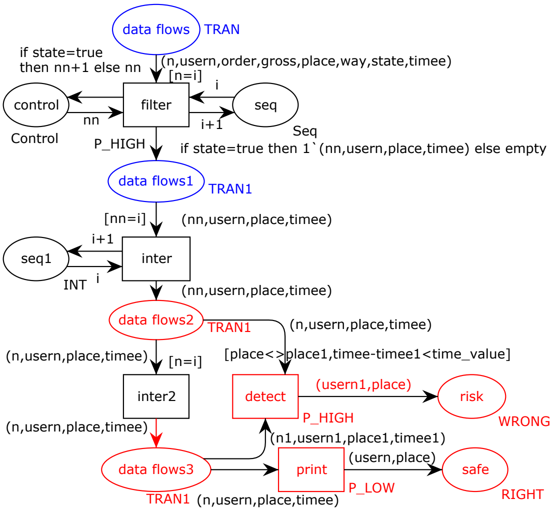


Fig. 3. The Event Pattern model for the anomaly detection of user's account address.

INT. The filtering conditions are represented by the output arc functions of "filter" and the transition's guard function $n = i$. The tokens of "data flows1" are used to identify the users' abnormal account addresses. The analysis operation of the next user behavior stream is controlled by the arc function *i + 1* of the place "seq".

In the identification phase, we define risky behaviors as *colset WRONG = product Usern * Address*, and risk-free behaviors as *colset RIGHT = product Usern * Address*. The guard function *place<>place1, timee-timee1<time_value* of the transition "detect" indicates that the user account address is abnormal in a short time. Since the priority of "detect" is higher than "inter2", if the tokens of "data flow2" and "data flow3" satisfy the guard function, "detect" can be fired, and the tokens of the place "risk" are risky behaviors. Otherwise, the transition "print" is fired, the tokens of the place "safe" are risk-free behaviors, indicating that the user's account is safe.

The anomaly detection of the user's account address mainly focuses on the address and time in the user's shopping data. If the user's physical address changes in a short time, we define the captured data stream as a risk behavior. The model structure of abnormal detection of user's payment way is the same as that of abnormal detection of user's account address, and it only needs to change the variables of the colored set and arc variable in the specific process, so it is not necessary to be introduced in detail in this paper.

3.1.2 Case 2 - The Event Pattern model for the anomaly detection of the user's payment amount

In this case, the abnormal payment amount means that the user places orders continuously and the purchase amount continues to increase. The model is shown in Fig. 4. The model is divided into three parts: the behavior filtering, the behavior anomaly judgment, and the judgment of payment amount continuously increasing. The blue area represents the behavior filtering phase, which is the same as Fig. 3. The only difference is that variable m is added to the color set of the place "data flow". The number of identification of user behaviors is determined by m, whose default value is 1. The red area represents the abnormal judgment of behaviors. The purple area is used to determine whether the payment amount continues to increase. In the model, we set the default number of identifications as 4.

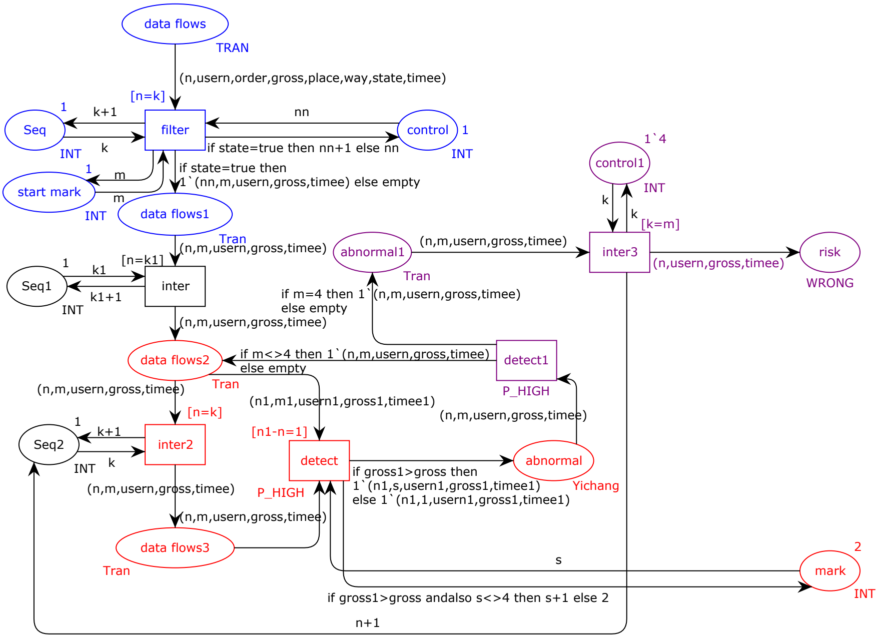


Fig. 4. The Event Pattern model for the anomaly detection of the continuous increase of user payment amount.

The tokens in the place "data flow" is used to simulate the data of user behaviors. The guard functions on the transitions ("filter", "inter", and "inter1") are used to control the firing sequence. When all the prepositive places of "filter" have tokens, it is enabled. The arc function *if state=true then nn+1 else nn* is used for renaming the behavior flow of whose payment state is already paid, so as to avoid the confusion of the identification order of the behavior flows, guarantee the accuracy of the identification result, and complete the filtering operation. Since the priority of transition “detect” is higher than that of “inter2”, once the places “data flow” and “data flow2” have token, which satisfies the conditions of the identification operation, the transition “detect” can be fired. At this point, the place “mark” will increase the number of times of identifications by 1. Once the identification is completed, the lowest-numbered token in “data flow” will enter in “data flow2” by firing the transition, and then a new round of identification will start. If the identification times of one user satisfy the marking of “control1”, “inter3” can be fired, and the token in “risk” indicates that the user account is at risk.

3.1.3 Case 3 - The Event Pattern model for the anomaly detection of the muti-accounts

The risky behaviors of the multi-account user refer to that the user has two or more accounts to place orders. Order characteristics include: the time interval among different orders are very short, and the order numbers are very similar, the payment statuses are different. Once the above characteristics are satisfied, the user’s behavior is judged as risk. The specific model is shown in Fig. 5. In the model, the place “timeshold” represents the time threshold. The characteristics of the user’s risk behaviors are mainly reflected in the guard function of the transition “detect”, and the arc function between the transition “detect” and the place “risk”. We can use the model to identify the illegal behaviors.

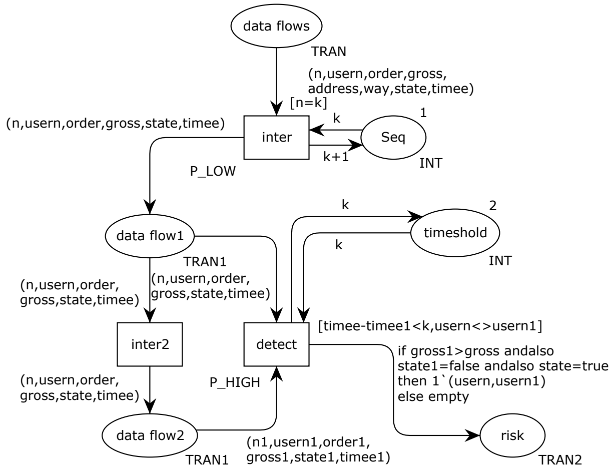


Fig. 5. The Event Pattern model of the anomaly detection of the muti-accounts.

3.2 Model validation

This section illustrates the verifying process by Algorithm 1. In Algorithm 1, the variable “Result” depicts whether Event Pattern model is correct. The value of “Result” is obtained by generating a state space diagram or state space report in CPN Tools. The state space diagram is a directed graph. Each reachable mark has a node. The state space report contains the bounded properties of the places. When we consider the reachable markings, the bounded properties can get the specific information.

Algorithm 1: The validation process

```
Input: ($CPN$, $M_0$), where $CPN$=($P, T, A, \Sigma, V, C, G, E, I$) is a Event
        Pattern model and $M_0$ a marking of it.
Output: Result.
Result=‘Correct’;
Generate a state space diagram or state space report;
if The number of leaf nodes in the state space diagram is >1 and there are
  many different markings then
    Result=‘False’;
else
    return Result;
end
if Lower in the state space report! = 0 then
    Result=‘False’;
else
    return Result;
end
```

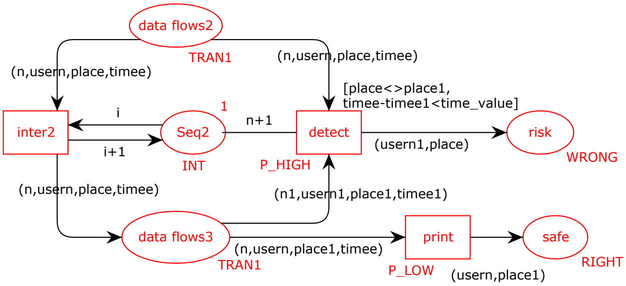


Fig. 6. The correct model structure.

By Algorithm 1, only the Event Pattern model of the user’s account address has errors. The reason is that when the place (“data flows1”) has a token or some tokens, the transition “inter2” can be fired, which results in the change of the order of the user shopping data flows. To solve this problem, we need to modify the transition's guard functions, meanwhile ensuring that the user data flows are executed sequentially. We modify the model as shown in Fig. 6.

3.3 Model to EPL

CEP refers to the real-time processing of all input event streams according to predefined event processing rules or Event Patterns. Once the input event streams meet the event processing rules, complex events will be generated. In this paper, the event processing rules are provided by Esper and described in the Event Pattern language, which is a like_SQL language [35,36].

After the modeling and validating of the Event Pattern model, the correctness is guaranteed. Then, the CPN models should be transformed to EPL according to the follow steps.

```
Algorithm 2: The generation of EPL
  Input: An Event Pattern model $CPN = (P, T, A, \Sigma, V, C, G, E, I)$.
  Output: An EPL.
  1. Translating $CPN$ to a simplified model $PN = (P, T, A)$;
  2. Make $PN$ as a directed graph, and the node access sequence $\phi = \{p_1, p_2, ...p_m\}$ is generated according to the depth traversal algorithm, where $m$ is the number of places;
  3. Generate the input matrix $A1_{m \times n}$ and output matrix $A2_{m \times n}$ of $PN$, $n$ is the number of the transitions;
  4. Using $\phi$, $A1_{m \times n}$ and $A2_{m \times n}$ to achieve structure matching, and generate the keywords: 'select', 'having', 'where', 'group by', 'from', and etc..;
  5. Variables are generated by the color set of the places, input and output variables of transitions, and guard functions;
  6. Generate expressions of clause based on the corresponding places, transitions, keywords;
  7. Obtain the EPL.
```

In the process of identifying the risky behaviors, the event type of the users' shopping behavior is defined as "TranEvent". The event attributes and the meanings are shown in Table 2. We have modeled the cases of possible risky Even Patterns in above section, and then we convert them into EPLs by Algorithm 2. The corresponding Event Patterns and EPLs as shown in Table 3.

Table 2. The event type of users' shopping behavior.

Variable name	**Attribute**	**Meaning**
userID	int	ID number of a user
gross	double	The amount paid
orderID	int	Order number
tradeway	string	The payment way
tradeplace	string	The payment place
tradetime	timestamp	The payment time
tradestate	bool	The Payment status

4 USER BEHAVIOR RISK IDENTIFICATION SYSTEM BASED ON ESPER

Nowadays, engines for CEP include Esper [1], Apache Flink [2], Oracle CEP [3], and etc.. Considering the open source characteristics of Esper and its EPL is more in line with the research in this paper, Esper is chosen to be the engine of user behavior risk identification system.

Esper supports real-time analysis and processing of massive event streams, which is done primarily through the JAVA [37]. The modular design of user behavior risk identification system is shown in Fig. 7, which mainly includes:

- Capture the behavior flow of online shopping users: serve as the data source of CEP;

- Input adapter is used to convert the captured data source into an event source through byte filtering, aggregation, etc.;

- Historical access database provides an interface to access the database. While processing the data in the window mechanism, the engine directly call the user behavior flow in the historical access;

- Esper engine is the core of CEP technology including acquiring the configurations, defining the events, defining the EPLs, defining listeners and binding listeners, etc.;

[1] http://www.espertech.com/esper/
[2] https://flink.apache.org/
[3] https://docs.oracle.com/

Table 3. The corresponding event patterns and EPLs.

No	Event Pattern	EPL
1	The user's account address changes within a short period of time (10s).	select * from pattern [every temp1 = TranEvent→temp2=TranEvent(temp2 .tradeplace!=temp1.tradeplace, temp1 .userID=temp2.userID, temp1.tradeway= temp2.tradeway,(temp2.tradetime. getTime()- temp1.tradetime.getTime()) /1000 <10, temp1.tradestate=true, temp1.tradestate =temp2.tradestate)]
2	The user's shopping way changes within a short period of time (10s).	select * from pattern [every temp1= TranEvent→temp2=TranEvent (temp2.tradeway!=temp1.tradeway, temp1.tradeplace=temp2.tradeplace, temp1.userID=temp2.userID, (temp2 .tradetime.getTime()-temp1 .tradetime.getTime())/1000 <10, temp1.tradestate=true,temp1.tradestate =temp2.tradestate)]
3	The user places orders continuously and the payment status of the orders are paid, the order amount presents an increasing trend and the last order amount is much lager than the first order amount.	select * from TranEvent match_recognize (measures A as temp1, B as temp2, C as temp3, D as temp4 pattern (A B C D) define A as A.gross IS NOT NULL,B as (A.gross < B.gross), C as (B.gross< C.gross), D as (C.gross<D.gross) and D.gross > 4*A.gross) having temp1.userID = temp2.userID and temp2.userID =temp3.userID and temp3.userID =temp4.userID and temp1.tradestate=1 and temp2.tradestate=1 and temp3 .tradestate=1 and temp4.tradestate=1
4	The people uses different accounts to purchase the same goods, and the former pays much more than the latter, it is worth noting that the former state of payment is unpaid, the latter state of payment is paid.	select * from pattern[every temp1= TranEvent→ temp2=TranEvent (temp2.tradeway=temp1.tradeway, temp1.tradeplace=temp2.traeplace, temp1.userID!=temp2.userID, (tmep2.tradetime.getTime()- temp1.tradetime.getTime())/1000 <5, temp1.tradestate!=temp2.tradestate)]

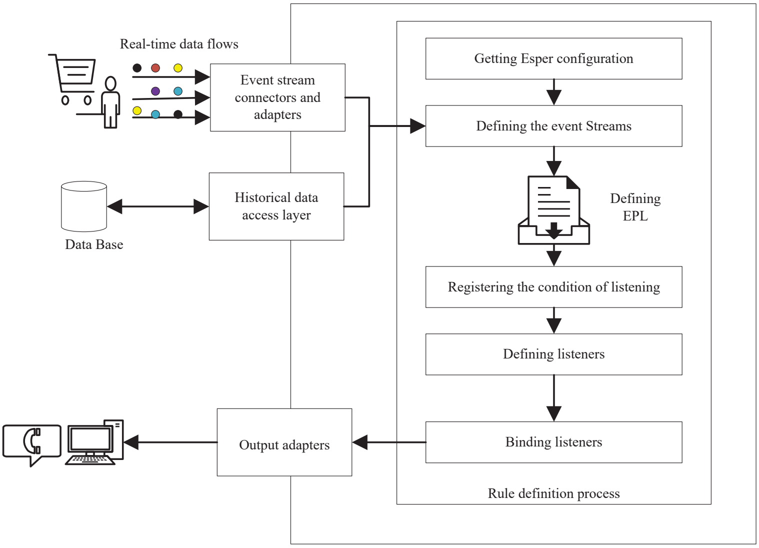

Real-time data flows
Event stream connectors and adapters
Historical data access layer
Data Base
Output adapters
Getting Esper configuration
Defining the event Streams
Defining EPL
Registering the condition of listening
Defining listeners
Binding listeners
Rule definition process

Fig. 7. The system architecture.

- Output adapter is used to send messages that the Esper engine is listening to external systems.

  Among them, Esper engine is mainly composed of the following steps:

- Get the configuration of the Esper engine:
  Configuration config=new Configuration();
  config.addEventType("TranEvent", TranEvent.class.getName());
  EPServiceProvider cep=EPServiceProviderManager.
  getProvider("myCEPEngine", config);
- Defining event types:
  When defining the event types, they usually includes JavaBean, Map, and XML. In order to facilitate analysis, we use JavaBean to define user behavior events for online shopping (TranEvent).
- Add EPL:
  String epl="select * from TranEvent.win:time(30 sec) group by usern";
- Register the condition of listening:
  EPAdministrator cepAm=cep.getEPAdministrator();
  EPStatement statement=cepAm.creaeEPL(epl);
- Defining listeners:
  The listener is an interface provided by Esper to listen for predefined rules in the engine. The listener is notified as soon as the event satisfies the EPL, because the interface contains the method *update*(), which involves two parameters, newEvents, and oldEventEvents, for receiving the events. At the same time, both

parameters are an array of event beans, and the method to get the field values in the EPL is eventbean.get("usern");

- Binding listeners:
  MyListener listener=new MyListener();
  Statement.addListener(listener).

As we cannot get the real trading data, we develop a demo online shopping platform (Fig. 8) to produce the simulation data. We produce the order information of simulation users from the specified date 2019.12.01 to 2019.12.30 on the demo platform. For example, the shopping information of the user (id=37983443) in this period is shown in Table 4. It can be clearly seen from the table that the payment address of orders no. 3 and no. 4 had changed within a short time, indicating that the user's account was at risk. We get all the order information of the user (id=38975436) in the period as shown in Table 5. During the specified period from 2019.12.23 13:30:23 to 2019.12.23 13:45:00, the user had placed orders continuously and the order amount had continued to increase, and the amount of the last order had been much larger than the amount of the first order, indicating that the user account was at risk. The order information of all users from 2019.12.24 08:00:00 to 2019.12.24 09:00:00 is shown in Table 6. We can see that:

1) The payment time between "18976512" and "19001416" is relatively short;

2) The products purchased by "18976512" and "19001416" are of the same type;

3) The order payment status of "18976512" is unpaid, while the order payment status of "19001416" is paid;

4) The payment amount of "18976512" is much larger than the latter.

These characters indicate that a user uses different accounts to place orders, which may lead to the risk of order replacement attack, so that the user account is at risk. Above examples matches cases 1-3. We input the shopping data streams into the Esper, and can seen from the Figs. 9(a)-(c) that the system can successfully capture the risky behaviors.

5 CONCLUSION

Nowadays, online shopping is an indispensable consumption way for people, which has a great impact on people's life. However, there are some risks in the process of online shopping. Real-time identification of online shopping user behaviors based on CEP can effectively avoid the generation of risk behaviors. In this work, we propose an Event Pattern modeling method based on CPN, which can depict and validate

Table 4. Online shopping user behavior flow(id=37983443)

NO	Gross	Order number	Payment way	Payment place	Payment time	Payment status
1	123	1	TPP1	Beijing	2019.12.12 19:30:23	true
2	56.8	2	TPP1	Beijing	2019.12.12 19:45:23	true
3	196	1	TPP1	Beijing	2019.12.17 08:30:23	true
4	444	2	TPP1	Xi'an	2019.12.17 08:30:28	true
5	156	1	TPP2	Beijing	2019.12.25 13:13:56	true

Table 5. Online shopping user behavior flow(id=38975436)

NO	Gross	Order number	Payment way	Payment place	Payment time	Payment status
1	456	1	TPP2	Xi'an	2019.12.01 19:30:23	true
2	325	1	TPP3	Xi'an	2019.12.17 15:30:28	false
3	356	1	TPP1	Beijing	2019.12.23 13:30:23	true
4	778	2	TPP1	Beijing	2019.12.23 13:35:11	true
5	779	3	TPP1	Xi'an	2019.12.23 13:39:45	true
6	19	4	TPP2	Xi'an	2019.12.23 13:43:09	false
7	1467	5	TPP1	Xi'an	2019.12.23 13:45:09	true

Table 6. User behavior flows in specified period

User Account	Gross	Payment way	Payment place	Payment time	Payment status	Product types
14537817	456	TPP3	Chongqing	2019.12.24 08:01:12	true	Home appliances
18976512	7980	TPP1	Beijing	2019.12.24 08:11:12	false	Gold ware
19001416	70	TPP1	Beijing	2019.12.24 08:11:36	true	Gold ware
34516537	779	TPP1	Nanjing	2019.12.24 08:30:54	true	Clothing

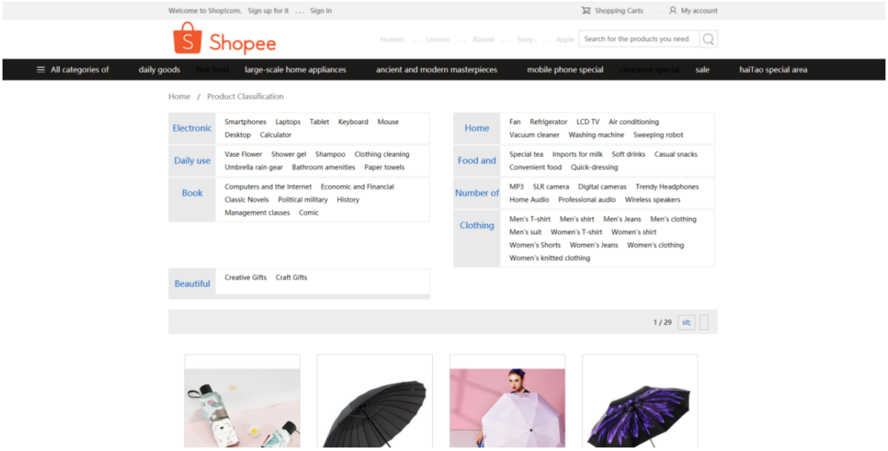
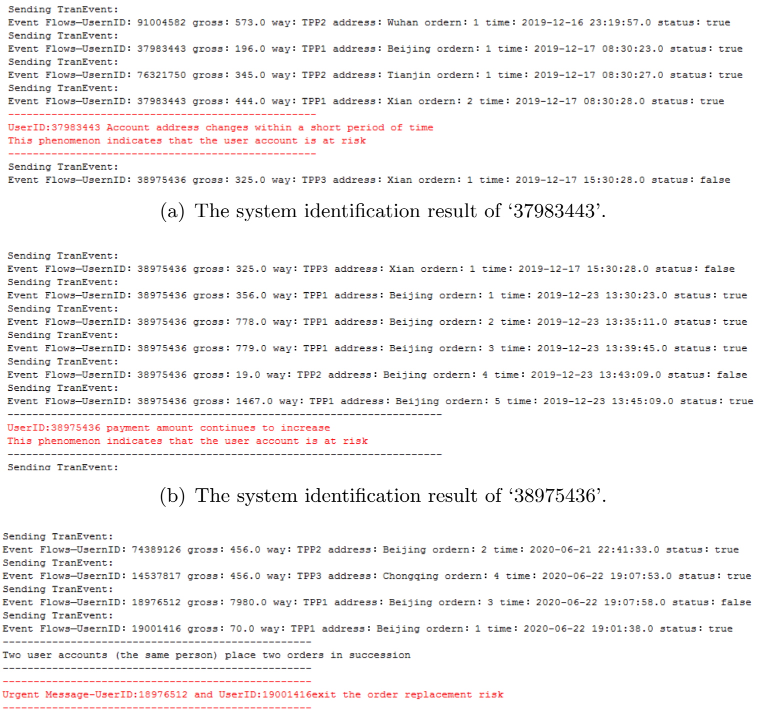

Welcome to Shop!com. Sign up for it ... Sign in
Shopping Carts
My account
Shopee
Huawei . .. Lenovo ... Xiaomi ... Sony . .. Apple
Search for the products you need
All categories of
daily goods
fine food
large-scale home appliances
ancient and modern masterpieces
mobile phone special
clearance special
sale
haiTao special area
Home / Product Classification
Electronic
Smartphones Laptops Tablet Keyboard Mouse
Desktop Calculator
Daily use
Vase Flower Shower gel Shampoo Clothing cleaning
Umbrella rain gear Bathroom amenities Paper towels
Book
Computers and the Internet Economic and Financial
Classic Novels Political military History
Management classes Comic
Beautiful
Creative Gifts Craft Gifts
Home
Fan Refrigerator LCD TV Air conditioning
Vacuum cleaner Washing machine Sweeping robot
Food and
Special tea Imports for milk Soft drinks Casual snacks
Convenient food Quick-dressing
Number of
MP3 SLR camera Digital cameras Trendy Headphones
Home Audio Professional audio Wireless speakers
Clothing
Men's T-shirt Men's shirt Men's Jeans Men's clothing
Men's suit Women's T-shirt Women's shirt
Women's Shorts Women's Jeans Women's clothing
Women's knitted clothing
1 / 29
slt;

Fig. 8. The demo online shopping platform.



```
Sending TranEvent:
Event Flows—UsernID: 91004582 gross: 573.0 way: TPP2 address: Wuhan ordern: 1 time: 2019-12-16 23:19:57.0 status: true
Sending TranEvent:
Event Flows—UsernID: 37983443 gross: 196.0 way: TPP1 address: Beijing ordern: 1 time: 2019-12-17 08:30:23.0 status: true
Sending TranEvent:
Event Flows—UsernID: 76321750 gross: 345.0 way: TPP2 address: Tianjin ordern: 1 time: 2019-12-17 08:30:27.0 status: true
Sending TranEvent:
Event Flows—UsernID: 37983443 gross: 444.0 way: TPP1 address: Xian ordern: 2 time: 2019-12-17 08:30:28.0 status: true
--------------------------------------------------
UserID:37983443 Account address changes within a short period of time
This phenomenon indicates that the user account is at risk
--------------------------------------------------
Sending TranEvent:
Event Flows—UsernID: 38975436 gross: 325.0 way: TPP3 address: Xian ordern: 1 time: 2019-12-17 15:30:28.0 status: false
```

(a) The system identification result of '37983443'.

```
Sending TranEvent:
Event Flows—UsernID: 38975436 gross: 325.0 way: TPP3 address: Xian ordern: 1 time: 2019-12-17 15:30:28.0 status: false
Sending TranEvent:
Event Flows—UsernID: 38975436 gross: 356.0 way: TPP1 address: Beijing ordern: 1 time: 2019-12-23 13:30:23.0 status: true
Sending TranEvent:
Event Flows—UsernID: 38975436 gross: 778.0 way: TPP1 address: Beijing ordern: 2 time: 2019-12-23 13:35:11.0 status: true
Sending TranEvent:
Event Flows—UsernID: 38975436 gross: 779.0 way: TPP1 address: Beijing ordern: 3 time: 2019-12-23 13:39:45.0 status: true
Sending TranEvent:
Event Flows—UsernID: 38975436 gross: 19.0 way: TPP2 address: Beijing ordern: 4 time: 2019-12-23 13:43:09.0 status: false
Sending TranEvent:
Event Flows—UsernID: 38975436 gross: 1467.0 way: TPP1 address: Beijing ordern: 5 time: 2019-12-23 13:45:09.0 status: true
-----------------------------------------------------------------------
UserID:38975436 payment amount continues to increase
This phenomenon indicates that the user account is at risk
-----------------------------------------------------------------------
Sending TranEvent:
```

(b) The system identification result of '38975436'.

```
Sending TranEvent:
Event Flows—UsernID: 74389126 gross: 456.0 way: TPP2 address: Beijing ordern: 2 time: 2020-06-21 22:41:33.0 status: true
Sending TranEvent:
Event Flows—UsernID: 14537817 gross: 456.0 way: TPP3 address: Chongqing ordern: 4 time: 2020-06-22 19:07:53.0 status: true
Sending TranEvent:
Event Flows—UsernID: 18976512 gross: 7980.0 way: TPP1 address: Beijing ordern: 3 time: 2020-06-22 19:07:58.0 status: false
Sending TranEvent:
Event Flows—UsernID: 19001416 gross: 70.0 way: TPP1 address: Beijing ordern: 1 time: 2020-06-22 19:01:38.0 status: true
--------------------------------------------------
Two user accounts (the same person) place two orders in succession
--------------------------------------------------
--------------------------------------------------
Urgent Message-UserID:18976512 and UserID:19001416exit the order replacement risk
--------------------------------------------------
```

(c) The system identification result of 'multi-account'.

Fig. 9. The system identification results

Event Patterns of user behavior risk effectively. Then, we convert the model to EPL according to the specific steps. Combining CEP with CPN to identify user behavior risk can improve the security of online shopping. In the future work, we will model risk behaviors of online shopping processes from the perspectives of consumers, sellers, and the third parties.

6 ACKNOWLEDGMENT

This work was supported by the Natural Science Foundation of Shaanxi Province under (No. 2021JM- 205) and Open Research Fund of Anhui Province Engineering Laboratory for Big Data Analysis and Early Warning Technology of Coal Mine Safety (No. CSBD2022-ZD05).

REFERENCES

[1] Z. Li, M. Huang, G. Liu, and C. Jiang, "A hybrid method with dynamic weighted entropy for handling the problem of class imbalance with overlap in credit card fraud detection," *Expert Systems with Applications*, vol. 175, p. 114750, 2021.

[2] E. Y. Chen, S. Chen, S. Qadeer, and R. Wang, "Securing multiparty online services via certification of symbolic transactions," in *Security and Privacy (SP), 2015 IEEE Symposium on.* IEEE, 2015, pp. 833–849.

[3] Y. Xie, G. Liu, C. Yan, C. Jiang, and M. Zhou, "Time-aware attention-based gated network for credit card fraud detection by extracting transactional behaviors," *IEEE Transactions on Computational Social Systems*, 2022.

[4] M. A. Sadikin and R. W. Wardhani, "Implementation of rsa 2048-bit and aes 256-bit with digital signature for secure electronic health record application," in *2016 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 2016, pp. 387–392.

[5] L. Wu, T. Chen, C. Qiao, and Z. Li, "Authentication technology of mobile internet of things based on the dynamic password," in *2018 IEEE International Conference of Safety Produce Informatization (IICSPI)*, 2018, pp. 204–208.

[6] A. Muñoz, J. Toutouh, and F. Jaime, *A Review of Dynamic Verification of Security and Dependability Properties*, 01 2019, pp. 162–187.

[7] M. Gull and A. Pervaiz, "Customer behavior analysis towards online shopping using data mining," in *International Multi-Topic ICT Conference.*

[8] M. E. Phillips, N. D. Stepp, J. Cruz-Albrecht, V. D. Sapio, and V. Sritapan, "Neuromorphic and early warning behavior-based authentication for mobile devices," in *2016 IEEE Symposium on Technologies for Homeland Security (HST)*, 2016.

[9] S. Wen, Y. Xue, J. Xu, L.-Y. Yuan, W.-L. Song, H.-J. Yang, and G.-N. Si, "Lom: Discovering logic flaws within mongodb-based web applications," *International Journal of Automation and Computing*, vol. 14, no. 1, pp. 106–118, 2017.

[10] F. Sun, L. Xu, and Z. Su, "Detecting logic vulnerabilities in e-commerce applications." in *NDSS*, 2014.

[11] R. Wang, S. Chen, X. Wang, and S. Qadeer, "How to shop for free online–security analysis of cashier-as-a-service based web stores," in *Security and Privacy (SP), 2011 IEEE Symposium on.* IEEE, 2011, pp. 465–480.

[12] C. W. Enumeration, "Cwe-840 business logic errors," 2014.

[13] C. Cwe, "472: External control of assumed-immutable web parameter."

[14] C. Jiang, J. Song, G. Liu, L. Zheng, and W. Luan, "Credit card fraud detection: A novel approach using aggregation strategy and feedback mechanism," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3637–3647, Oct 2018.

[15] L. Zheng, G. Liu, C. Yan, C. Jiang, M. Zhou, and M. Li, "Improved tradaboost and its application to transaction fraud detection," *IEEE Transactions on Computational Social Systems*, vol. PP, pp. 1–13, 08 2020.

[16] Y. Xie, G. Liu, C. Yan, C. Jiang, M. Zhou, and M. Li, "Learning transactional behavioral representations for credit card fraud detection," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[17] W. Yu, Z. Ding, L. Liu, X. Wang, and R. D. Crossley, "Petri net-based methods for analyzing structural security in e-commerce business processes," *Future Generation Computer Systems*, vol. 109, pp. 611 – 620, 2020.

[18] L. Xing, Y. Chen, X. Wang, and S. Chen, "Integuard: Toward automatic protection of third-party web service integrations." in *NDSS*, 2013.

[19] W. Yu, C. Yan, Z. Ding, C. Jiang, and M. Zhou, "Analyzing e-commerce business process nets via incidence matrix and reduction," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 1, pp. 130–141, 2018.

[20] G. Cugola and A. Margara, "Processing flows of information: From data stream to complex event processing," *ACM Computing Surveys*, vol. 44, no. 3, 2012.

[21] D. Bonino and L. De Russis, "Complex event processing for city officers: A filter and pipe visual approach," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 775–783, 2018.

[22] M. Weidlich, H. Ziekow, A. Gal, J. Mendling, and M. Weske, "Optimizing event pattern matching using business process models," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 11, pp. 2759–2773, 2014.

[23] Z. Milosevic, A. Berry, W. Chen, and F. A. Rabhi, "An event-based model to support distributed real-time analytics: Finance case study," in *2015 IEEE 19th International Enterprise Distributed Object Computing Conference*, 2015, pp. 122–127.

[24] Y. Liu and D. Wang, "Complex event processing engine for large volume of rfid data," in *2010 Second International Workshop on Education Technology and Computer Science*, vol. 1, 2010, pp. 429–432.

[25] R. Bhargavi, V. Vaidehi, P. T. V. Bhuvaneswari, P. Balamurali, and G. Chandra, "Complex event processing for object tracking in wireless sensor networks," in *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, vol. 3, 2010, pp. 211–214.

[26] Z. Ma, W. Yu, X. Zhai, and M. Jia, "A complex event processing-based online shopping user risk identification system," *IEEE Access*, vol. 7, pp. 172 088–172 096, 2019.

[27] K. Jensen, *Coloured Petri nets: basic concepts, analysis methods and practical use.* Springer Science & Business Media, 2013, vol. 1.

[28] W. Yu, M. Jia, X. Fang, Y. Lu, and J. Xu, "Modeling and analysis of medical resource allocation based on timed colored petri net," *Future Generation Computer Systems*, vol. 111, pp. 368 – 374, 2020.

[29] W. M. van der Aalst, N. Lohmann, and M. La Rosa, "Ensuring correctness during process configuration via partner synthesis," *Information Systems*, vol. 37, no. 6, pp. 574–592, 2012.

[30] Y. Du, X. Li, and P. Xiong, "A petri net approach to mediation-aided composition of web services," *IEEE Transactions on Automation Science and Engineering*, vol. 9, no. 2, pp. 429–435, 2012.

[31] H. Hu and Y. Liu, "Supervisor simplification for ams based on petri nets and inequality analysis," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 1, pp. 66–77, 2014.

[32] S. Wang, C. Wang, and M. Zhou, "Design of optimal monitor-based supervisors for a class of petri nets with uncontrollable transitions," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 5, pp. 1248–1255, 2013.

[33] H. Macià, V. Valero, G. Díaz, J. Boubeta-Puig, and G. Ortiz, "Complex event processing modeling by prioritized colored petri nets," *IEEE Access*, vol. 4, pp. 7425–7439, 2016.

[34] K. Jensen and L. M. Kristensen, *Coloured Petri nets: modelling and validation of concurrent systems.* Springer Science & Business Media, 2009.

[35] W. Ding, H. Wang, P. Nan, Y. Xiao, and Z. Liu, "Stock technical analysis system based on real-time stream processing," in *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, 2017.

[36] S. Sta, M. Lindeberg, and V. Goebel, "Online analysis of myocardial ischemia from medical sensor data streams with esper," in *Proceedings of the First International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL 2008)*, 2008.

[37] A. Mathew, "Benchmarking of complex event processing engine-esper," *Dept. Comput. Sci. Eng., Indian Inst. Technol. Bombay, Maharashtra, India, Tech. Rep. IITB/CSE/2014/April/61*, 2014.