A Multi-perspective Fraud Detection Method for Multi-Participant E-commerce Transactions

WangYang Yu, YaDi Wang, Lu Liu, YiSheng An, Bo Yuan, and John Panneerselvam

***Abstract*—Detection and prevention of fraudulent transactions in e-commerce platforms have always been the focus of transaction security systems. However, due to the concealment of e-commerce, it is not easy to capture attackers solely based on the historic order information. Many researches try to develop technologies to prevent the frauds, which have not considered the dynamic behaviors of users from multiple perspectives. This leads to an inefficient detection of fraudulent behaviors. To this end, this paper proposes a novel fraud detection method that integrates machine-learning and process mining models to monitor real-time user behaviors. First, we establish a process model concerning the B2C e-commerce platform, by incorporating the detection of user behaviors. Second, a method for analyzing abnormalities that can extract important features from event logs is presented. Then, we feed the extracted features to a Support Vector Machine (SVM) based classification model that can detect fraud behaviors. We demonstrate the effectiveness of our method in capturing dynamic fraudulent behaviors in e-commerce systems through the experiments.**

***keywords*—Fraud detection; Electronic transaction; Petri net; Machine learning**

I. INTRODUCTION

WITH the increasing popularity of e-commerce platforms, more and more commercial transactions are now relying on web-based systems than the traditional cash-based approach [1]. Although the entity economy is greatly impacted by the COVID-19 epidemic in recent years, e-commerce remains largely unaffected by the pandemic, whereby aiding a steady market growth [2]. The sales volume of B2C (Business to Customer) e-commerce is expected to reach 6.5 trillion dollars by 2023 [3].

Though the growth of e-commerce and the expansion of modern technologies offer better opportunities for online businesses, new security threats have emerged over the past few years. Reportedly, the significant increase in the number of online fraud cases costs billions of dollars worldwide every year [4]. The dynamic and distributed nature of the Internet has made anti-fraud systems inevitable to ensure the security of online transactions. Existing fraud detection systems focusing on detecting abnormal user behaviors still characterize vulnerabilities when mitigating emerging security threats. An important issue in existing fraud detection systems is their lack of efficient process management during the trading process. The imperfect monitoring function is one of the key issues that need attention [5]. The detection perspective is usually not enough due to the lack of process capture for the existing work. To this end, we propose a process-based method, where user behaviors are recorded and analyzed in real-time, and historical data is transformed into controllable data. In addition, we incorporate a multi-perspective detection of abnormal behaviors.

This paper combines the advantages of process mining and machine learning models by introducing a hybrid method to solve the anomaly detection in data flows, which provides information about each action embedded in a control flow model. By modeling and analyzing the business process of the e-commerce system, this method can dynamically detect changes in user behaviors, transaction processes, and non-compliance situations, and comprehensively analyze and identify fraudulent transactions from multiple perspectives. Important contributions of this paper are listed as follows:

1) A conformance checking method based on process mining is applied in the field of e-commerce transactions to capture the abnormalities.
2) A user behavior detection method is proposed to perform comprehensive anomaly detection based on Petri nets.
3) An SVM model is developed by embedding a multi-perspective process mining into machine learning methods to automatically classify fraudulent behaviors.

The rest of this paper is organized as follows: Section 2 introduces the related work. Section 3 presents a model analysis and a background study. Section 4 forms the theoretical basis and describes our proposed fraud detection method. Section 5 presents and discusses the results of our experiments and Section 6 validates our proposed fraud detection method. Section 7 concludes our paper along with outlining our future research directions.

II. RELATED WORK

Existing fraud detection methods are categorized into non-formal approaches such as machine learning, and formal approaches such as process mining.

The machine-learning-based methods learn from previously obtained historical data to perform classifications

This work is supported in part by the Natural Science Foundation of Shaanxi Province under Grants 2021JM-205, National Natural Science Foundation of China under Grant 52172325, and in part by the fundamental research funds for the central universities under Grant 300102242902. (*Corresponding Authors*: *YaDi Wang and Lu Liu*).

W. Yu and Y. Wang are with the Key Laboratory of Modern Teaching Technology, Ministry of Education, Xi'an 710062, China, and also with the School of Computer Science, Shaanxi Normal University, Xi'an 710119, China (E-mail: ywy191@snnu.edu.cn and wyd@snnu.edu.cn).

L. Liu, B Yuan and J Panneerselvam are with the School of Computing and Mathematical Sciences, University of Leicester, Leicester LE1 7RH, U.K. (E-mail: l.liu@leicester.ac.uk, b.yuan@leicester.ac.uk and j.panneerselvam@leicester.ac.uk).

Y. An is with the School of Information Engineering, Chang'an University, Xi'an, China (E-mail: aysm@chd.edu.cn).

Manuscript received ***, 2022; revised ***, 2022.

or predictions of future observations to identify potential risky offline or online transactions [6]. Xuetong Niu et al. conducted a comparative study on credit card fraud detection methods that rely on machine-learning algorithms. Most of the machine-learning models perform well on the dataset of credit card transactions. Moreover, supervised models perform slightly better than unsupervised models after additional pre-processing, such as removing outliers [7].

Credit card fraud detection is widely deployed at the application layer, which uses the idea of discovering specific abnormal user behaviors to detect fraud. The supervised learning algorithm is the most commonly used learning method in online fraud monitoring transactions, since it has higher accuracy and coverage. Recent research in [8, 9] has proved that the machine learning method can efficiently capture fraudulent transactions in credit card applications. Fraudsters often change their behavioral pattern dynamically to overcome existing fraud detection methods. In online credit card fraud detection, SVM can classify user behaviors under complex scenarios and deliver reliable results [10]. Many researchers take the advantage of combining multiple detection methods for comprehensive fraud detection. For example, focusing on payment fraud applications, Dahee Choi et al. proposed a method by combining supervised and unsupervised learning [11]. Most of the machine learning-based methods use historical data to analyze fraudulent transactions. They have not given enough emphasis to the transactional process flow and dynamic user behaviors.

The second type of fraud detection methods uses process mining, focusing on extracting knowledge from existing event logs in information systems for the purpose of monitoring and improving the operational process in business IT infrastructure [12]. Process mining specializes in comparing the event log with an established model to further detect, locate, and interpret the deviation between the established model and the actual event log [13].

Process mining can detect a large number of abnormal transactions, which are not known to be identifiable by traditional methods. M Jans et al. postulated the emerging process mining approach as an appropriate solution to mitigate against fraud incorporating internal affairs [14]. For example, C Rinner et al. applied conformance checks to monitor the process of melanoma patients [15]. Asare et al. applied alignment and replay to check the conformance of the electronic medical record log and the hospital workflow model [16]. Research has focused on monitoring and evaluating the sequence of processes occurring in the historical medical event log by establishing corresponding training and testing models for conformance checking [17]. Tools such as ProM, Disco and Heustic miner are largely used for conformance checking. Process mining can be an efficient approach for fraud detection.

Especially, it is important to be dynamic and multi-perspective when detecting fraudulent user behaviors [18]. Process mining helps to compare the actual data against the standard model to identify outliers. Despite existing progress in fraud detection, it is still necessary to develop hybrid-learning methods to improve the accuracy of detection [19]. To promote the understanding and development of process mining for anomaly detection, a method of multi-perspective anomaly detection is proposed that goes beyond the perspective of control flow including time and resources [20]. Febriyanti et al. [21] assumed any noticeable changes in business processes as a suspected fraud behavior and proposed a method to detect some suspicious abnormal behaviors using a hybrid method of association rules and process mining. Previous research on using process mining to detect fraudulent transactions showed that process mining is capable of detecting fraudulent transactions, and it can effectively prevent audit fraud at a much earlier stage due to the continuous monitoring nature of event logs [22].

In conclusion, many of the existing machine learning methods only consider static user behaviors based on their occurrence rate. Only a very few studies have investigated real-time, dynamic, and multi-perspective factors of user behaviors in the e-commerce transaction process, which offers great control of the entire transaction process. The detection system based on process mining can record and analyze the changes in user behaviors and their preferences on time. However, analysis of complex details increases the number of variables or factors that should be considered, which makes the detection model more complex.

III. MODEL ANALYSIS

An e-commerce platform is an information interaction platform that provides online transactions for enterprises and/or individuals. The coverage rate of B2C (Business to Customer) e-commerce platforms is higher than that of other e-commerce platforms, and B2C has become the mainstream model of e-commerce in China [23]. The recent market trend of e-commerce has given emergence to various types of electronic payment systems. Third-party payment platforms supervise and restrict both the buyers and merchants within the terms of the transaction, thereby ensuring the legitimate rights and interests of both buyers and sellers. The process of e-commerce transactions is abstracted and the process flow is established as follow.

A. Process analysis

In a typical B2C process, buyers, e-commerce platforms, and sellers interact with each other. As shown in Fig. 1, the electronic transaction process encompasses five different participants including *Seller*, *Buyer*, the third-party cashier *TP*, the B2C trading platform *BCS* (Buyer and Seller Server), and the cashier server *CS*. This paper summarizes the transaction payment process as follows:

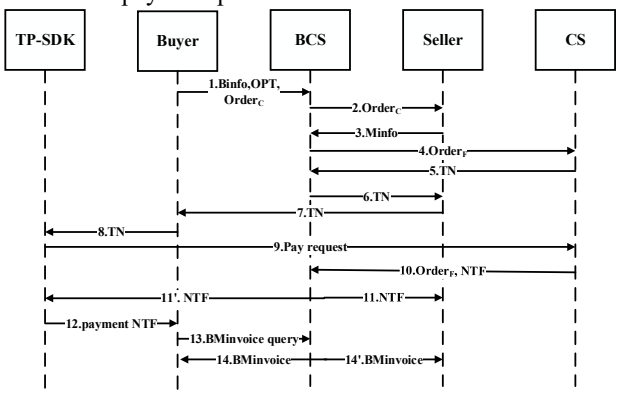


Fig. 1. Interaction flow of the transaction process.

1) After the *Buyer* logs in, the *Buyer* performs a series of operations on the user client device to purchase goods or services. The *BCS* generates a commodity order $order_C$ according to the products or services that are purchased. Commodity order $order_C$ is then passed to the Seller through the e-commerce platform.
2) The Seller makes a decision based on the information in $order_C$, and the Seller passes on the willingness to the platform *BCS*. If the order is rejected, the *Buyer* returns to the user operation process; if the order is accepted, the system establishes a pre-payment formal order $order_F$, which contains the detailed information purchase of the user.
3) After the payment is completed, *CS* signs the formal order $order_F$. Then *CS* sends two payment completion notifications *NTF*, of which one is to notify *TP*, and the other is to notify the seller. The buyer's click triggers the *UpdateOrderStatus* function, in other words, the order status is updated. Afterward, the paid order information $order_F$' is generated.
4) The *Seller* checks the order invoice with the payment status. After that, the *BCS* checks the order and current transaction details.
5) To notify the *CS* of the upcoming payment, *CS* generates a unique transaction number (TN) of the payment information, and then *CS* passes this transaction code to the platform server *BCS*.
6) After the *Seller* receives the TN, it signs and passes TN to the buyer client. At this time, the *Buyer* can confirm the order payment information and enters the password, or cancels the order. Then, the *Buyer* requests payment and enters a password. If the password is correct, the process proceeds to the next step. Otherwise, the transaction fails.
7) The third-party payment client *TP* processes the request, and verifies the credit score and signature of the user. If it is normal, the *TP* makes the payment and sends the payment request command to the *CS*.

B. Fraud mode analysis

To capture the fraudulent behaviors effectively, we define some common fraud modes [23][24] and abstract them as follows.

1) Fraud mode one - an order is tempered by a malicious actor:
   The malicious actor may deceive the victim merchant by sending a fake formal payment order $order_F^A$ to the cashier server. The malicious actor obtained the order items that do not match the payment value by tampering with the order information, such as the total amount.
2) Fraud mode two - subcontract the order:
   The victim pays the malicious actor's order instead of his order. To achieve their goals, the malicious actors impersonate the duties of sellers and buyers. The order information changes before and after the payment.
3) Fraud mode three - send fake notifications:
   During such attacks, the malicious actor submits an order instead of paying for the order, but sends a fake payment result notification to notify the seller that the order is successfully paid.
4) Fraud mode four - paying a cheap order to get expensive goods:
   First, the malicious actor submits a cheap order as an ordinary buyer, and then submits an expensive order but does not pay. However, the system marks the order as "pending". The malicious actor replaces the paid order with the current order at this time.

IV. ANALYTICAL METHODS

Fig. 2 depicts the framework of our detection method. Firstly, the transaction event log is filtered and cleaned, and a database of user behavior mode is constructed in the data preparation stage. Secondly, we perform an analysis on the control flow, resource, throughput time analysis, data flow, and user behavior on the event logs, and extract the abnormalities of each transition from different perspectives as the training features of fraud detection. Then machine-learning algorithms are implemented, and finally, an SVM model is built to classify fraudulent transactions. Our proposed fraud detection method is introduced in detail as follows.

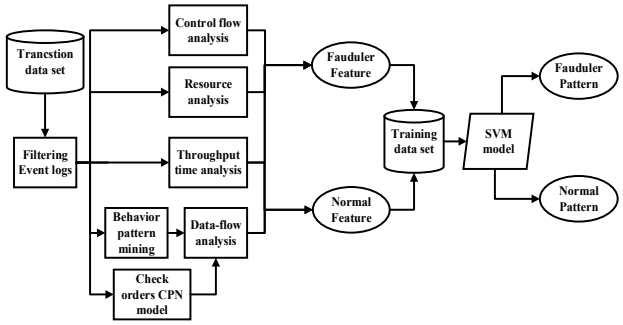


Fig. 2. The proposed framework.

A. Theoretical basis

An event log is made of multiple traces. Each trace represents the life cycle of one case [25], which is specifically composed of case, event, timestamp, action, and resource.

This section introduces establishing the link between the current action, which is shown as an event log, and the action of the model. When some real-life event log is replayed on the process model, some transitions are introduced for routing purposes rather than representing the actual work [26]. We only consider actions of practical significance using a Labeled Petri net defined as follows.

Definition 1. System Net [26]

SN is a system net $SN=(lPN, M_{init}, M_{final})$, $lPN=(P, T, F, l)$ is a Petri net with a labeling function, U_A is defined as universe of action labels, then the label can be formally defined as: $l=T\nrightarrow U_A$; M_{init} is the initial markings and M_{final} is the final markings, which are the tokens contained within the markings of the Petri net.

Definition 2. Data Petri net $DPN=(SN, V, U, R, W, G)$ in which:

- *SN* is a system net $SN=(lPN, M_{init}, M_{final})$ based on $lPN=(P, T, F, l)$;
- V is a set of data variables that are used in the transitions;
- U is a function that defines the range of each value, i.e., D_v is the domain of variable values v, and the value of all variables must be within the range defined. For each value $v \in V$, $U(v)=D_v$;
- R is a read function $R \in T \rightarrow \rho(V)$, which indicates the sets of variables that should be read for each transition;
- W is a write function $W \in T \rightarrow \rho(W)$, which indicates the sets of variables that should be written for each transition;
- G is a guard function $G \in T \rightarrow (V_W \cup V_R)$, which is represented by some combination rules of reading variables and writing variables such that for any transition $t \in T$, and for any variables $v \in V$, if v_r in $G(t)$, then $v \in R(t)$, for any variables $v \in V$, if v_w in $G(t)$, then $v \in W(t)$;

$(DPN, (M, s))[b> (M', s')$ describes an enabled binding b in marking (M, s) may occur. The result is the marking (M', s') after the occurrence. It represents the transition of a net system from one state to another. In *DPN*, the new transitions after triggering should update the newly written variables to all variable sets, i.e., $s_{new} = s \oplus w$, in where $s_{new}(v) = w(v)$ for all $v \in write(t)$, and $s_{new}(v) = s(v)$ for all $v \in V \backslash write(t)$.

Definition 3. Trace and event logs [26]

U_{VN} is a universe of variable names, U_{VV} is the universe of variable values, and U_{VM} is the partial mapping from variable names to values, i.e., $U_{VM} = U_{VN} \nrightarrow U_{VV}$;

A trace, which is defined as a set of action sequences with input and output data, can be represented as $\delta \in (U_A \times U_{VM} \times U_{VM})^*$. In the same way, an event log is composed of multiple sets of traces, which can be expressed as $L \in \mathcal{B}((U_A \times U_{VM} \times U_{VM})^*)$.

Definition 4. Cost function with optimal alignment

s_M is the Data Petri net model, and s_L is the event log; γ is defined as the alignment result of s_M and s_L. In order to quantify the degree of deviation, a cost function is used to define the movements that exist in the above alignment results, i.e., $\kappa \in \Sigma \rightarrow R + 0$. For $\forall(s_L, s_M) \in \Sigma$, if s_L=>> or s_M = >>, then $K^{std}_{(s_L, s_M)} = 1$; otherwise, $K^{std}_{(s_L, s_M)} = 0$. The sequence cost is the sum of costs of individual moves in the sequence, i.e., $K(\gamma) = \sum_{(s_L, s_M) \in \gamma} \kappa(s_L, s_M)$. For all alignment results γ' of the event log and Data Petri net model, there is an optimal alignment $K(\gamma) \leq K(\gamma')$.

B. Multi-perspective conformance checking

After the rules are formally defined, conformance checking is used to detect abnormalities. Conformance checking requires an alignment of event log L and process model *DPN*, which is the alignment of each single trace $\delta \in L$ and the process model *DPN*.

The event log of the system records detailed information such as the occurrence time, executor, and interaction data in each action. Through conformance checking, some special trajectories that do not match the trajectories of commonly occurring actions are identified, so that anomalies can be initially detected in a single perspective. For some special cases, such as malicious actors fraudulently using legal accounts to conduct illegal operations or even fraudulent actions, comprehensive analysis and judgment should be carried out in combination with the inspection results from multiple perspectives. In this paper, any trace in the event log that does not conform to the model is suspected for potential anomalies, and the following definition is adopted from [26].

Definition 5. Deviations between the event log and process model

A set (*act*, *r*, *w*, *res*, *time*) is defined, where the read variable of the action *act* is *r*, the write variable is *w*, its resource attribute is represented by *res*, and the throughput time is represented by *time*. The traces in event logs are represented as $S_L = U_A \times U_{VM} \times U_{VM}$, and traces of Data Petri net can be represented as $S_{DPN} = T \times U_{VM} \times U_{VM}$. According to the definition in [26], ">>" means that there is no corresponding move. We use this definition to indicate occurrences of deviations. To replay the event log in the model, different types of deviations are defined as follows:

- Deviation only in log: $\{s_L=(l(t), r, w, res, time) \in S_L\} \cap \{s_M=>>\}$;
- Deviation only in *DPN* model: $\{s_M \in S_{DPN}\} \cap \{s_L=>>\}$;
- Deviation in both model and logs with correct data attributes: $\{s_M=(t, r, w) \in S_{DPN}\} \cap \{s_L=(l(t), r, w)\}$;
- Deviation in both model and logs with incorrect data attributes: $\{s_M=(t, r, w) \in S_{DPN}\} \cap \{s_L=(l(t), r_l, w_l)\} \cap \{s_L=(r \neq r_l \mid w \neq w_l)\}$;
- Deviation in resource: $s_L(res)$!= $s_M(res)$;
- Deviation in time: $s_L(time)$ = unqualified;
- All other deviations are considered as abnormal.

The identification of unqualified traces is valuable [25]. The focus of our analysis is to obtain a specific meaning of the points that do not conform to the guards, and information that is hidden in the abnormal points. For multiple control-flow alignments, the optimal alignments γ is selected. Fig. 3 shows a Petri model mined from a set of event log. Four deviations exist between traces in the event log and traces in the model, which are represented as grey areas in Table I. According to the path of Petri net model, from the perspective of control flow, the event log, t_0 has occurred twice. The only deviation in the event log means redundant actions. After the occurrence of t_3, there is t_1 rather than t_0, therefore the only deviation in the model representing some actions is skipped. The throughput time of action t_1 in the 5th line does not meet the threshold requirement. Action t_0 has a deviation in the resource, presented in the 6th line of Table I.

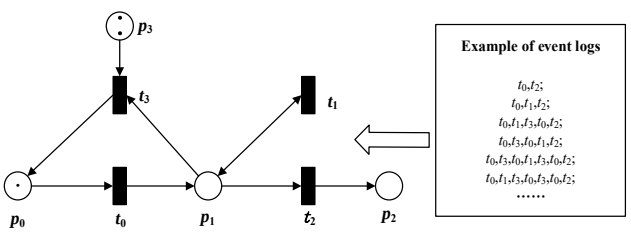


Fig. 3. The deviations example

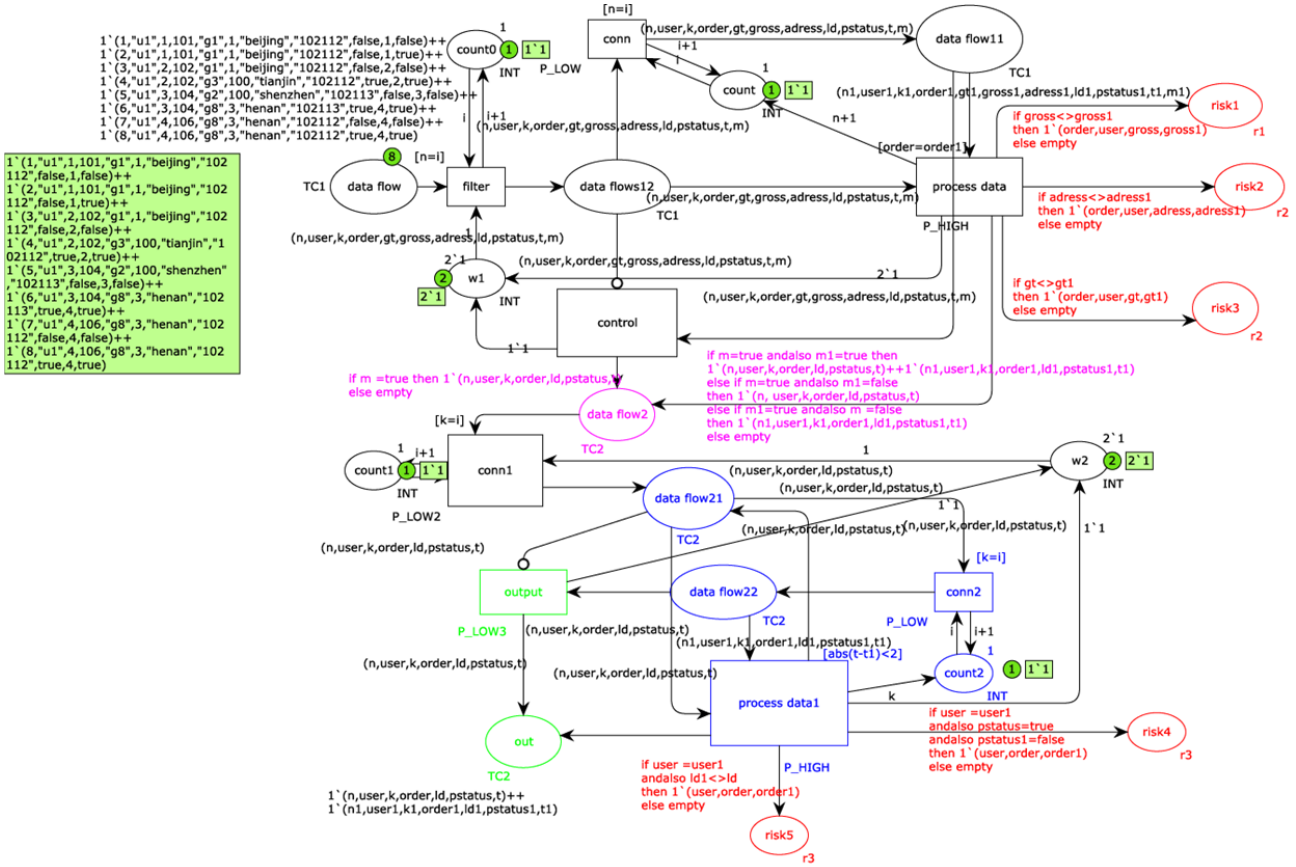


Fig. 4. CPN model: check and process orders.

TABLE I
DEVIATIONS EXAMPLE

	Event log traces	*Model traces*
1	(t_0, {(*att1*:3%), (*att2*:3000)}, {Ø}, {*resource*: Mike}, {*throughput-time*: qualified})	(t_0, {(*att1*:4%), (*att2*:3000)}, {Ø}, {*resource*: Michael}, {*throughput-time*: qualified})
2	(t_0, {(att1:4%), (att2:2000)}, {Ø}, {resource:Michael}, {throughput-time: qualified})	>>
3	(t_3, {(*att3*: true)}, {Ø}, {*resource*: Mike}, {*throughput-time*: qualified)	(t_3, {(*att3*: true)}, {Ø}, {*resource*: Mike}, {*throughput-time*: qualified)
4	>>	(t_0, {(*att1*:4%), (*att2*:3000)}, {Ø}, {*resource*: Michael}, {*throughput-time*: qualified})
5	(t_1, {(*att4*: VIP)}, {Ø}, {*resource*: Kris}, {*throughput-time*: unqualified)	(t_1, {(*att4*: VIP)}, {Ø}, {*resource*: Kris}, {throughput-time: qualified)
6	(t_2, {(*att5*:3)}, {Ø}, {*resource*: Mike}, {*throughput-time*: qualified)	(t_2, {(*att5*:3)}, {Ø}, {*resource*: Amber}, {*throughput-time*: qualified)

C. Check and process orders

Colored Petri net (CPN) [27] is a powerful modeling tool for concurrent and distributed systems, which can not only be used to process and analyze users' transaction orders, but to realize the formalization and visualization of the detection process dynamically. The CPN model in Fig. 4 corresponds to the actions *Check and Process Orders* in the business process model. The CPN model is established for detecting and processing order information according to the detection target of actions. The detection target of the CPN model mainly includes: (1) detecting whether there are other unpaid orders under the same user ID in a short period of time; (2) detecting whether the IP address of the order under the same user ID has changed in a short period of time; (3) detecting whether the amount in commodity orders is consistent with the actual payment amount of the final order for the same order from the same user; (4) detecting whether the address in commodity orders is consistent with the actual address of the final order; and (5) detecting whether the goods type in commodity orders is consistent with the actual goods type of the final order.

As shown in Fig. 4, the place "*data flow*11" and the place "*data flow*12" in the CPN model realize the comparison of commodity orders and final transaction orders of the same transaction one by one. The place "*w*1" controls the number and order of detection. The transition "*process data*" can compare the commodity order with the final one. The places "*process data*" aims to find the orders with unusual order information, when its rules set by the arc function are satisfied, the token is the input to the corresponding places "*risk*1", "*risk*2", and "*risk*3" respectively. Among them, the existence of the token in the place "*risk*1" represents the abnormal change in the order amount. These tokens represent order information, such that abnormal orders can be visually observed and extracted.

Before detecting whether a given user has other unpaid orders, the order information flow should be filtered first. This is because the object of this type of anomaly detection is the final orders of the transaction, and our input data flow contains two types of order information. As shown in Fig. 4, the pink part of the model shows the filtering function used to obtain the final transaction orders. Next, the transition

"*process data*1" processes the order data. If the arc function of "*risk*4" is satisfied, a new token is generated in the place "*risk*4"; in the same way, if the arc function pointing to "*risk*5" is satisfied, i.e., a new token is generated in the place "*risk*5". Abnormal order information can now be extracted from the tokens. In summary, the CPN model realizes the functions of processing and detecting order information.

D. User operation behavior detection

Next, we add the data flow perspective based on the above detection method, which integrates the function of user behavior detection. Buyer behavior analysis can be divided into two parts: static attribute and dynamic behavior [28].

The user's static attribute data used in this paper mainly includes IP address, login time, and operation duration. We use the Apriori algorithm [29] to obtain the normal patterns based on the user's historical static attribute data. Before using operational data for mode mining, static attributes should be pre-processed and described using mathematical models. The login time is expressed as an integer of [0, 24]. Table II shows the characteristics of static attributes.

TABLE II
USER STATIC ATTRIBUTE PRE-PROCESSING

Attribute	*IP address*	*Login time*	*Operation duration (min)*
Example	192.168.1.249	21	60

When the user operates on the APP or web browser, a series of operation data is generated; the user's behavior habits are hidden in these data. When other users use the same device, account, and IP address as the actual user, the behavioral patterns obtained from the user's historical behavioral data are used for pattern matching. As shown in Table III, the user behavior data used in this method mainly includes searching for products, browsing products, favorites, adding to shopping carts, viewing shopping carts, and so on. The categories of behavior data are limited, so that integers are used to label and classify user behaviors.

TABLE III
USER BEHAVIOR CATEGORIES AND MATHEMATICAL IDENTIFICATION

Behavior category	*Symbol*
Search	0
Browser	1
Favorite	2
Add to the cart	3
View the cart	4
View the favorites	5
Submit the orders	6

By identifying the temporal logic relationship rules through data mining algorithms, user behavior patterns are obtained. This paper uses the GSP algorithm [30] to mine user behavior data.

Fig. 5 shows the specific steps of user anomaly detection. Pattern matching of the user's operation behavior corresponds to the functions in transition D and U, which are used to detect and analyze the operation and static attributes. (*act*, *att*, *att'*, *alg*) represents the input variable of action *act* is *att*, and the new variable obtained by the algorithm *alg* is *att'*. For example, a set of bindings for dynamic behavior habits can be represented as (*D*, *OPT_type*, *Operation*, *recursive_correlation_ function*). The similarity of action *D* is obtained after processing using the recursive correlation function [31]. A predefined threshold is used to determine whether the current user behavior is abnormal or not. Similarly, a set of valid bindings for static attributes can be represented as (*U*, *Static_User_Pattern*, *Static_att, Full_sequence_comparison*). It indicates that the full sequence comparison method [31] is used to compare the current static attribute with the static attribute pattern. Combined with the process, the matching similarity *Static_att* is used as the *w* of the action, and the judgment can be made according to the threshold value.

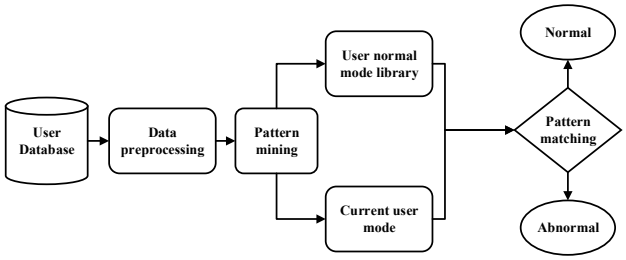


Fig. 5. User data pattern mining and detection process.

E. An anomaly extraction algorithm for e-commerce business process based on multi-perspective

Through the analysis and modeling of the e-commerce transaction process structure mentioned above, the reference model of the e-commerce business process based on Petri net is obtained. The inputs of Algorithm 1 include the e-commerce business process model, the CPN model, and the sequence of actions that occur in real-time. Firstly, the event log and the reference model are optimally aligned. Then according to steps 2.1, 2.2, 2.3, and 2.4, the anomalies of the current transaction's event trajectory in the control flow perspective, resource flow perspective, time flow perspective, and data flow perspective are calculated respectively, and the abnormal point sequence $B_d[T_x]$ corresponding to the abnormal point sequence $B_d[T_x]$ is recorded as 1. The abnormal point sequence $B_d[T_x]$ corresponding to the transition T_x without offset is recorded as 0. Step 3 counts the results of processing orders; Steps 4 and 5 count the results of user static attributes and dynamic behavior detection, and record them in the corresponding abnormal point sequence $B_d[T_x]$. The final algorithm output result is the abnormal point set $B[T_x]$, which is used as the initial data of SVM model training for fraud detection.

Algorithm 1: The anomaly extraction algorithm for e-commerce transaction process based on multi-perspectives

Input:
Business process model S_{DPN}= (P, T, F, V, U, R, W, G); CPN model of check and process orders; Event logs S_L.

Output:
The sequence of anomalies for SL is stored in B[t_x], $t_x \in T$, x is the serial number, x∈[0, 32].

//Step1 :Initialize parameters.

1 $B=\varnothing$, where $B[t_x]=B_c[t_x]\cup B_r[t_x]\cup B_t[t_x]\cup B_d[t_x]$ respectively represent the set of anomalies of control flow, resource flow, time flow, and data flow that may occur in transition t_x, and the judgment result of user behavior is placed in the set $B_d[t_x]$.

//Step2 : According to the process mining conformance checking algorithm, the optimal alignment result is denoted as γ, which

can be judged as follows.
if $\{s_L=(l(t_x), r, w, res, time)\in S_L\} \cap \{s_M=>>\}$ or $\{s_M\in S_{DPN}\}\cap \{s_L=>>\}$, **then**
  $B_c[t_x]=1$
Else
  $B_c[t_x]=0$
end if
if $s_L(res)$!= $s_M(res)$, **then**
  $B_r[t_x]=1$
Else
  $B_r[t_x]=0$
end if
if $s_L(time)$ = *unqualified*, **then**
  $B_i[t_x]=1$
Else
  $B_i[t_x]=0$
end if
if $\{\{s_M=(t_x, r, w)\in S_{DPN}\}\cap\{s_L=(l(t_x), r_l, w_l)\in S_L\} \cap \{r\neq r_l \mid w\neq w_l\}\}\cup\{s_L(\text{Guards}(t_x)) = False\}$, **then**
  $B_d[t_x]=1$
Else
  $B_d[t_x]=0$
end if

// Step3: For the detection result of order information, the function of CPN model corresponds to transition t_{28} in *DPN* model, the event log S_L is input into the model to obtain the result.
if there exists *token* in the end places $risk_i$, $i\in[1,5]$, **then**
  the abnormal conditions corresponding to $risk_i$ of the end places with *token* are recorded in the sequence $B_d[t_x^i]$: $risk_i \rightarrow B_d[t_{28}^i]=1$
Else
  $risk_i \rightarrow B_d[t_{28}^i]=0$
end if
if the static attribute of the user does not meet the threshold, **then**
  $B_d[t_{29}]=1$
Else
  $B_d[t_{29}]=0$
end if
if the user dynamic behavior does not meet the threshold, **then**
  $B_d[t_4]=1$
Else
  $B_d[t_4]=0$
end if
Return $B[t_x]$

F. Fraud detection based on SVM

The evaluation of a single perspective is relatively one-sided and cannot accurately determine whether the current transaction is fraudulent or not. Therefore, it is very important to integrate the detection results of each perspective to evaluate the transaction's status as a whole. Next, the multi-perspective detection results are used as the features, and SVM is used to learn from these features and to integrate them for evaluating whether the current transaction has fraudulent behavior or not as a whole.

1) Classification problem and SVM [32]

The problem of fraud detection is essentially a binary classification problem, which can be solved by a classification model. The binary classification problem is a process in which a classification function judges whether the input data belongs to the positive class or the negative class. The mathematical definition is as follows:

$$h(x) = p(y=1 \mid x), y = 0 \text{ or } 1 \tag{1}$$

where, x presents the input data; y presents the class of the input data; $h(x)$ is the classification function.

Anomaly detection is a process in which a detection model uses user data to judge whether the user is abnormal. Anomaly detection satisfies the definition of the classification problem. An SVM model is a supervised learning method that can be used to solve binary classification problems. Compared with other classification methods, SVM delivers better performance with less sample size. In addition, SVM is good at using the kernel function to solve the case where the data is linearly inseparable.

The key role of SVM is to find a suitable hyperplane to divide samples into two classes, and maximize the distance between the samples and the hyperplane. The loss function of SVM is as follows:

$$loss = \sum_{i=1}^{N} \max\left(0, 1-y_i\left(\omega^T x_i + b\right)\right) + \lambda\|\omega\|^2 \tag{2}$$

where, x_i is the feature vector of the i-th sample; y_i is the label of the i-th sample; ω is the weight parameter; b is the bias parameter; λ is the regularization coefficient.

Through learning from the dataset and updating the weights, the loss function of the SVM model gradually decreases and finally converges. After the above process, the SVM model is successfully constructed and used for prediction. By taking the features of the current user as input, the SVM model classifies whether the current transaction behaviors are fraudulent or not.

2) Feature selection for anomaly detection

In the process of multi-perspective detection, each perspective gives an inference about whether the current transaction has fraudulent behaviors or not. The SVM model takes the detection inference of these perspectives as features. We obtain 82 anomaly detection features from the Data Petri net and data mining process. These features are used to detect whether a current transaction is abnormal from multiple perspectives. These features are used as the feature vectors in the SVM model. Parts of features and their meanings are shown in Table IV. These features are respectively the control flow analysis results of 20 actions, the time flow analysis results of 20 actions, the resource flow analysis results of 20 actions, and the data flow analysis results of 22 actions.

TABLE IV
EXAMPLES OF FEATURES

Feature	*Feature name*	*Meaning*
X_1	Control flow analysis result of transition A	Whether the control flow of transition A is abnormal
X_2	Control flow analysis result of transition B	Whether the control flow of transition B is abnormal
X_{27}	Time flow analysis result of transition H	Whether the time flow of transition H is abnormal
X_{45}	Resource analysis result of transition F	Whether the resource flow of transition F is abnormal
X_{62}	Data flow analysis result of transition D	Whether the user's operation behavior is abnormal
X_{82}	Data flow analysis result of transition U	Whether the user's static attributes are abnormal

V. PROCESS MINING EXPERIMENT RESULTS

This paper uses the process-mining tool ProM LIte 1.2 as the experimental platform [33]. Data flow experiments and fusing multi-view experiments use Python3.7 and the machine learning framework Scikit-Learn-0.22.

A. Control flow analysis

According to our proposed method introduced in the previous section, we generate the control flow, as shown in Fig. 6.

Fig. 6. Part of control flow analysis results.

This section uses the plug-in *Replay a Log on Petri Net for Conformance Analysis* to complete the control flow detection. In Fig. 6, we intercepted several traces in the result. The green part represents "move both on log and model", which is normal. The grey part means "move on the model only" which depicts that the event log has no deviations corresponding to the model. The purple part means "move on a log only", which means that the model has no deviation corresponding to the event log, further indicating that the event log is abnormal, that is, skipped actions. For example, actions 5 and 6 are skipped in trace66. This always means that the order placement represented in this trace is not approved by the merchant.

B. Throughput time analysis

Throughput time is the interval among actions, which can be obtained by analyzing the completion time among actions recorded in the event log. We use the plug-in Replay a Log for Performance/Conformance Analysis for the throughput time analysis [34]. Fig. 7 depicts the time interval among each action. The time interval of each action in the actual e-commerce process is very close, and the time difference is in the order of microseconds. For the sake of intuition, this paper extends the running time of each action. We set the lowest threshold of the transition processing time to 10 milliseconds, and the highest threshold to 60 seconds.

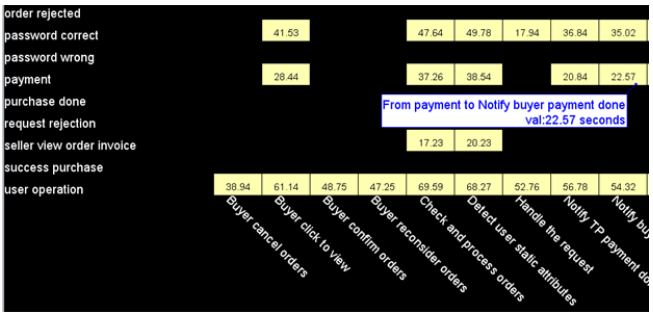


Fig. 7. Partial results of throughput time analysis.

C. Resource analysis

This section analyzes each participant, who performed the actions recorded in the event log. We used the plug-in Multi Perspective Explorer [35] from ProM to detect the outliers in the resource.

In Fig. 8, different actions are completed by different performers and visually displayed in different colors in the analysis panel of ProM. For example, light blue represents the Buyer, and dark blue means that the executor is BCS. The purple flag means that an unauthorized participant has performed the corresponding action. For example, for the action Order Created in the first trace, its executor is an unauthorized user.

D. Data flow analysis

Data flow analysis is introduced to address the shortcomings of control flow analysis. It provides information about each action embedded in the process model [18]. To get the results of data flow analysis, firstly we use the plug-in *Edit Petri Net with Data* to obtain the Data Petri net model. The complete results are shown in Fig. 9.

The guard is the normal threshold set for each action. Specifically, it is normal when the current input value meets the standards set by the guards of an action. When no guard meets the configurations of data flow, that action point is deemed as an abnormal data flow. Action Submit Orders sequence writes the ID and types of the product that the buyer wants to purchase. Fig. 10 is a snippet of this sequence. It can be seen that the input of the action *Order Addressed By Seller* is the goodsID and goodstype.

To arrive at a clearer result, the plug-in Multi-Perspective Process Explorer and Conformance Checking are used to match and analyze the event log and the *DPN*. The result is shown in Fig. 11, where each action is represented with different colors. For instance, green represents the move both on model and log, purple means move on the model only, and grey represents invisible actions, that is, skipped actions. By clicking on a given action, we can obtain the matching information between the model and the event log in the data flow of each action. The data marked in red indicates a mismatch. We extract these suspicious anomalies and use them as the basis for subsequent training using machine-learning models.

VI. EXPERIMENT AND ANALYSIS BASED ON SVM MODEL

This section utilizes the user anomaly detection features as data sources, which are obtained from multi-perspective detection and uses the SVM model to determine whether there exists any fraud. This experiment utilizes the grid search method [36] to adjust the hyper-parameters of the SVM model, the obtained hyper-parameters and the split ratio are chosen to perform a cross-validation experiment.

A. Data-set construction

Each data consists of 82 anomaly detection features, and each feature characterizes a value of 0 or 1, where 1 represents abnormality and 0 represents normality. A representation of the dataset used in our experiments is shown in Table V.

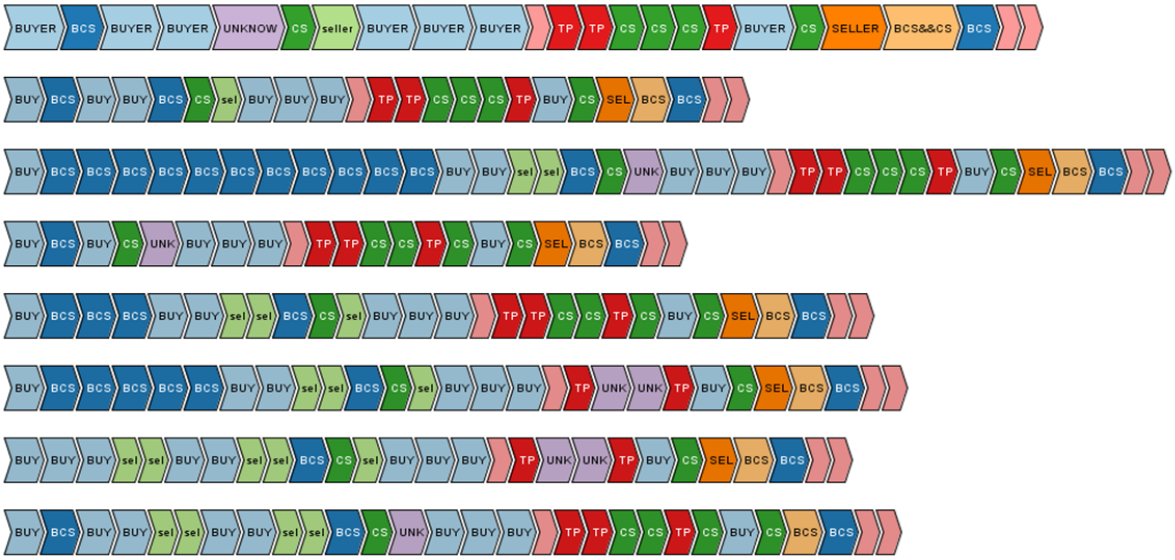


Fig. 8. Resource analysis by ProM.

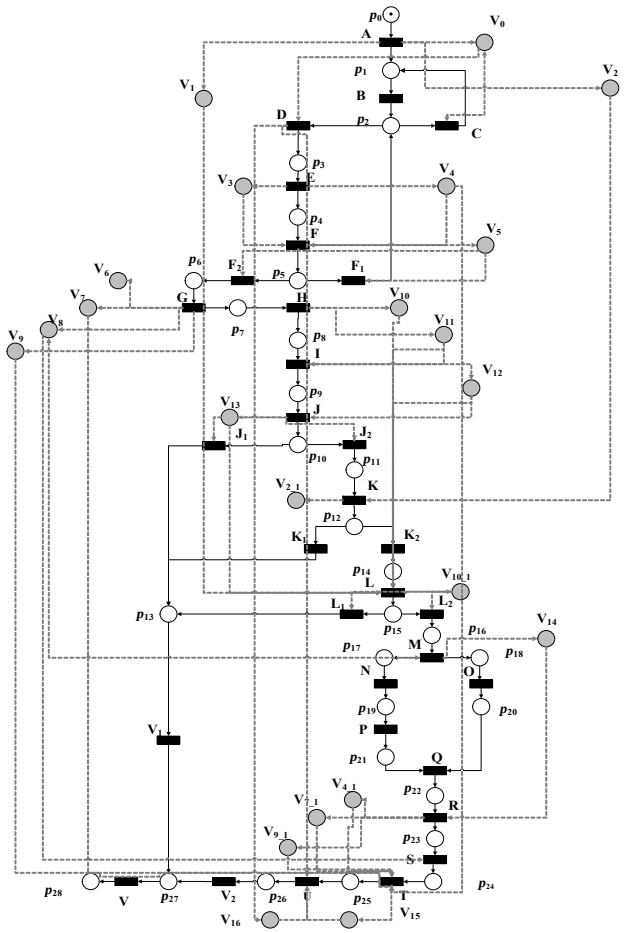


Fig. 9. Data Petri net model of the e-commerce transaction process.

According to Algorithm 1, the training data represented in Table V is obtained, the SVM model is trained by using the obtained dataset, and finally, an anomaly detection model is obtained. For example, in the first row of Table V, each control flow appears to be normal from every perspective, which is marked as 0, and the classification result is 0. In the second row, X_5 marked as 1 represents the action Order Addressed by Seller is skipped, X_{27} marked as 1 represents the execution time of action TN Created by Order takes too long, X_{45} marked as 1 represents the abnormal performer of the action Order Created, and X_{76} marked as 1 means the order payment amount is modified. Herein, this entire sequence is determined as the first fraud mode. In the third row, X_{11} and X_{12} marked as 1 indicate that the payment-related actions are skipped, X_{53} and X_{54} marked as 1 indicate that the performer of the notification action after the payment is abnormal, and X_{68} marked as 1 means that the user's credit level is low and does not meet the threshold. Thus, the entire sequence is determined as the third fraud mode.

TABLE V
EXAMPLES OF TRAINING DATA IN THE DATASET

Example1	*Marking*	*Example2*	*Marking*	*Example3*	*Marking*
X_1	0	…	0	…	0
X_2	0	X_5	1	X_{11}	1
…	0	…	0	X_{12}	1
X_{21}	0	X_{27}	1	…	0
X_{22}	0	…	0	X_{53}	1
X_{23}	0	X_{45}	1	X_{54}	1
X_{24}	0	…	0	…	0
…	0	X_{76}	1	X_{68}	1
X_{82}	0	…	0	…	0

B. Model performance test experiment based on K-fold cross validation method

This section verifies the validity of our SVM-based fraud detection model through comparative experiments. According to the data type, the experiment encompasses three scenarios: control flow characteristic data only, data flow characteristic data only, and both control flow characteristic data and data flow characteristic data. These three cases are tested by cross-validation method respectively to obtain the model precision (Precision), recall rate (Recall), F1-Score and AUC (Area under the ROC Curve) under the current data, whereby the model performance is analyzed according to the above indicators.

$$\begin{cases} precision = \dfrac{TP}{TP+FP} \\ recall = \dfrac{TP}{TP+FN} \\ F1 = \dfrac{2*precision*recall}{precision+recall} \end{cases} \quad (3)$$

TP represents the number of positive samples that are predicted as a positive class; *FP* represents the number of negative samples that are predicted as a positive class; *FN* represents the number of positive samples that are predicted as a negative class.

We use the grid search method to select the best hyper parameters of the SVM model, which is an enumeration method that is used to select the most optimal combination of hyper parameters. Through the grid search method, the optimal SVM hyper-parameters are obtained, as shown in Table VI.

TABLE VI
HYPER-PARAMETER SETTINGS OF THE SVM MODEL

Hyperparameter	*Meaning*	*Setting*
kernel	Kernel Function	Polynomial Kernel
C	Regularization	3
gamma	Kernel coefficient	0.25
degree	Highest degree of Polynomial Kernel	3
tol	Stop criterion	0.1

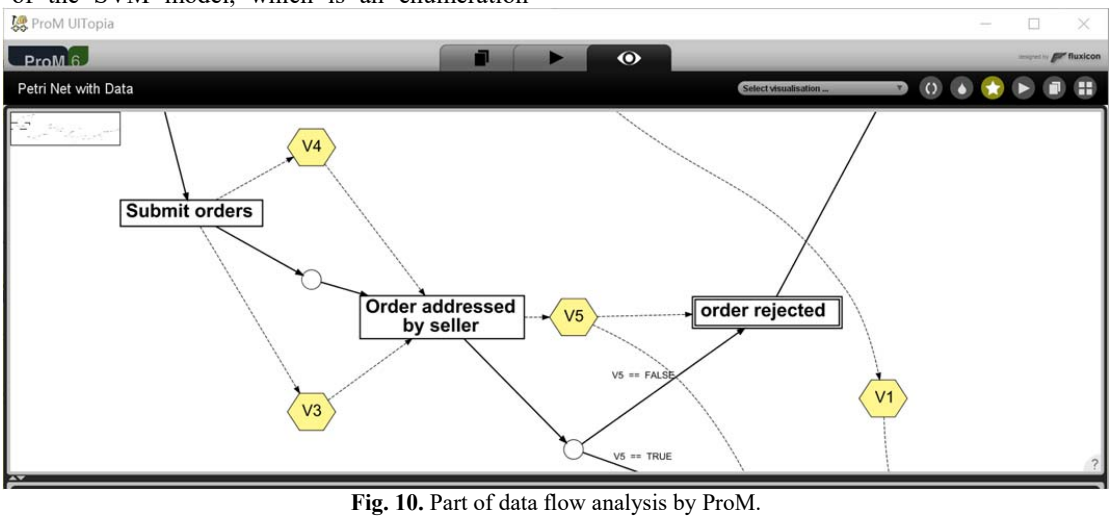


Fig. 10. Part of data flow analysis by ProM.

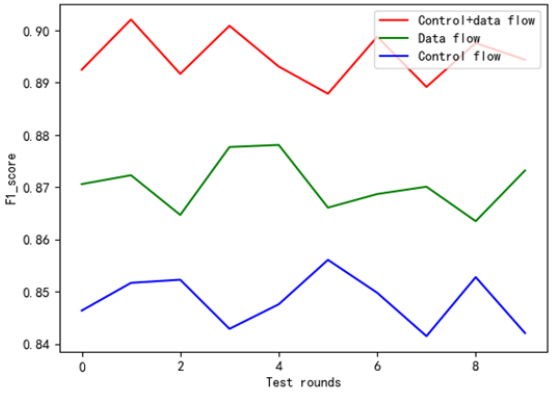
Fig. 11. The result of data flow analysis.

K fold Cross-validation [37] is an effective way of verifying the effectiveness of the model performance. In this experiment, the value of k is 10, that is, the experiment is carried out through 10 fold- cross verification.

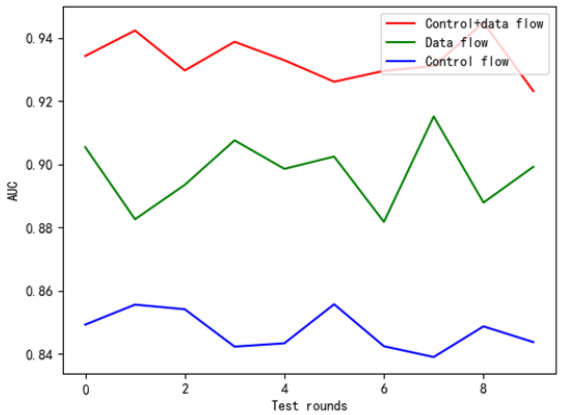
Fig. 12 (a) and (b) represents the statistical detection indicators of F1-core and AUC of our proposed SVM-based fraud detection model, obtained based on the 10-fold cross validation. Among them, the blue curve is the control flow index, the green curve is the model score considering the data flow, and the red curve represents the score under the fusion of multi-perspective features. As seen from Fig. 12, F1-score under the data fusion of control flow and data flow is higher than that when only one type of data is considered, that is, when the data of control flow and data flow are considered comprehensively, better user anomaly detection is obtained.

TABLE VII
FRAUD DETECTION MODEL RESULTS BASED ON SVM

Perspectives	*Precision*	*Recall*	*F1-score*	*AUC*
Control+data flow	0.946	0.852	0.895	0.935
Data flow	0.912	0.837	0.871	0.892
Control flow	0.889	0.812	0.849	0.842



(a) F1-score statistics



(b) AUC statistics

Fig. 12. F1-score and AUC statistics for three situations.

To further validate the fraud detection effects of our model under the three aforementioned cases, we consider various performance indicators under 50 rounds of tests and calculate their average values. The results are shown in Table VII. As seen from Table VII, the index of F1-score and AUC are both greater than indexes that consider only one of the perspectives under the case of integrating data flow and control flow features. These two kinds of characteristic data only consider one aspect of the user's anomaly. After learning the two types of data through the machine-learning model, the information of the two aspects is fully utilized to comprehensively detect user anomalies with better effect.

In summary, when compared with considering only one perspective of information, our proposed model characterizes a higher F1-score and AUC indicators. The detection effect of abnormal e-commerce users is better in our model. Therefore, our proposed method can detect abnormal e-commerce users more comprehensively. In addition, compared with the related deep learning methods for the fraud detection in e-commerce, our methodology can depict the transaction process and structures, and it is interpretable.

VII. CONCLUSION

This paper proposed a hybrid method to capture fraud transactions by integrating the formal process modeling and the dynamic user behaviors. We analyzed the e-commerce transaction process under five major perspectives: control flow perspective, resource perspective, time perspective, data perspective, and user behavior patterns. This paper utilized high-level Petri nets as the basis of process modeling to model the abnormal user behaviors and created an SVM model to perform fraudulent transaction detection. Our extensive experiments showed that the proposed method can effectively capture fraudulent transactions and behaviors. The overall index of our proposed multi-perspective detection method outperformed the single-perspective detection method. As our future work, related deep learning [38-42] and model checking methods [43-45] would be incorporated in the proposed framework for higher accuracy. Additionally, it's also a future work to incorporate more time features to the behavior patterns so as to make the risk identification more accurate. Furthermore, we will conduct research on constructing a standard fraud mode library, and apply the proposed methodology to other malicious behavior areas by coordinating the models.

REFERENCES

[1] R. A. Kuscu, Y. Cicekcisoy, and U. Bozoklu, *Electronic Payment Systems in Electronic Commerce*. Turkey: IGI Global, 2020, pp. 114–139.

[2] M. Abdelrhim, and A. Elsayed, "The Effect of COVID-19 Spread on the e-commerce market: The case of the 5 largest e-commerce companies in the world." *Available at SSRN 3621166*, 2020, doi: 10.2139/ssrn.3621166.

[3] P. Rao et al., "The e-commerce supply chain and environmental sustainability: An empirical investigation on the online retail sector." *Cogent. Bus. Manag.*, vol. 8, no. 1, pp. 1938377, 2021.

[4] S. D. Dhobe, K. K. Tighare, and S. S. Dake, "A review on prevention of fraud in electronic payment gateway using secret code," *Int. J. Res. Eng. Sci. Manag.*, vol. 3, no. 1, pp. 602-606, Jun. 2020.

[5] A. Abdallah, M. A. Maarof, and A. Zainal, "Fraud detection system: A survey," *J. Netw. Comput. Appl.*, vol. 68, pp. 90-113, Apr. 2016.

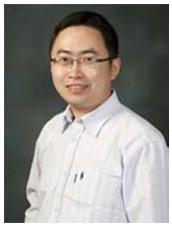
[6] E. A. Minastireanu, and G. Mesnita, "An Analysis of the Most Used Machine Learning Algorithms for Online Fraud Detection," *Info. Econ.*, vol. 23, no. 1, 2019.

[7] X. Niu, L. Wang, and X. Yang, "A comparison study of credit card fraud detection: Supervised versus unsupervised," *arXiv preprint arXiv*: vol. 1904, no. 10604, 2019, doi: 10.48550/arXiv.1904.10604.

[8] L. Zheng et al., "Transaction Fraud Detection Based on Total Order Relation and Behavior Diversity," *IEEE Trans. Computat. Social Syst.*, vol. 5, no. 3, pp. 796-806, 2018.
[9] Z. Li, G. Liu, and C. Jiang, "Deep Representation Learning With Full Center Loss for Credit Card Fraud Detection," *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 2, pp. 569-579, 2020.
[10] I. M. Mary, and M. Priyadharsini, "Online Transaction Fraud Detection System," in *2021 Int. Conf. Adv. C. Inno. Tech. Engr.* (*ICACITE*), 2021, pp. 14-16.
[11] D. Choi, and K. Lee, "Machine learning based approach to financial fraud detection process in mobile payment system," *IT Conv. P.* (*INPRA*), vol. 5, no. 4, pp. 12-24, 2017.
[12] R. Sarno et al., "Hybrid Association Rule Learning and Process Mining for Fraud Detection," *IAENG Int. J. C. Sci.*, vol. 42, no. 2, 2015.
[13] J. J. Stoop, "Process mining and fraud detection-A case study on the theoretical and practical value of using process mining for the detection of fraudulent behavior in the procurement process," M.S. thesis, Netherlands, ENS: University of Twente, 2012.
[14] M. Jans et al., "A business process mining application for internal transaction fraud mitigation," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 13351-13359, 2011.
[15] C. Rinner et al., "Process mining and conformance checking of long running processes in the context of melanoma surveillance," *Int. J. Env. Res. Pub. He.*, vol. 15, no. 12, pp. 2809, 2018.
[16] E. Asare, L. Wang, and X. Fang, "Conformance Checking: Workflow of Hospitals and Workflow of Open-Source EMRs," *IEEE Access*, vol. 8, pp. 139546-139566, 2020.
[17] W. Chomyat and W. Premchaiswadi, "Process mining on medical treatment history using conformance checking," in *2016 14th Int. Conf. ICT K. Eng.* (*ICT&KE*), 2016, pp. 77-83.
[18] M. D. Leoni, W. M. Van Der Aalst, and B. F. V. Dongen, "Data-and resource-aware conformance checking of business processes," in *Int. Conf. Bus. Info. Sys.*, Springer, Berlin, Heidelberg, 2012. pp. 48-59.
[19] S. M. Najem, and S. M. Kadeem, "A survey on fraud detection techniques in ecommerce," *Tech-Knowledge*, vol. 1, no. 1, pp. 33-47, 2021.
[20] K. Böhmer, and S. Rinderle-Ma, "Anomaly detection in business process runtime behavior--challenges and limitations," *arXiv preprint arXiv*, 2017, doi: 10.48550/arXiv.1705.06659.
[21] K. D. Febriyanti, R. Sarno and Y. Effendi, "Fraud detection on event logs using fuzzy association rule learning," in *2017 11th Int. Conf. Info. Comm. Tech. Sys.*, Surabaya, Indonesia, 2017, pp. 149-154.
[22] T. Chiu, Y. Wang and M. Vasarhelyi, "A framework of applying process mining for fraud scheme detection," *SSRN Electronic Journal*, 2017, doi:10.2139/ssrn.2995286.
[23] W. Yang et al., "Show Me the Money! Finding Flawed Implementations of Third-party In-app Payment in Android Apps," in *Proc. NDSS*, Shanghai, China, 2017.
[24] W. Rui, S. Chen, X. Wang and S.Qadeer, "How to Shop for Free Online--Security Analysis of Cashier-as-a-Service Based Web Stores," in *Proc. SSP*, Oakland, CA, USA, 2011, pp. 465-480.
[25] E. Ramezani, D. Fahland and W. Aalst, "Where did I misbehave? Diagnostic information in compliance checking," in *BPM.*, Berlin，Germany, Springer, 2012, pp. 262-278.
[26] M. Leoni, J. Munoz-Gama, J. Carmona and W. Aalst, "Decomposing alignment-based conformance checking of data-aware process models," in *Proc. OTM*, Amantea, Italy, 2014, pp. 3-20.
[27] K. Jensen, "Coloured Petri Nets: A High Level Language for System Design and Analysis," *DAIMI Report Series*, vol. 19, no. 338, pp. 342-416, Mar. 1993.
[28] B. Ji., H. Li., W. Han and Y. Jia, "Research on e-commerce-oriented user abnormal behaviour detection," *Netinfo Security*, Sep. 2014.
[29] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. VLDB*, S. F., USA, 1994, pp. 487-499.
[30] R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," in *Proc. EDBT*, Avignon, France, 1996, pp. 1-17.
[31] Y. Lian, Y. Dai and H. Wang, "Anomaly detection of user behaviors based on profile mining," *Chinese J. Computat-Ch.*, vol. 25, no. 3, pp. 325-330, Mar. 2002.
[32] C. Cortes and V Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp.273-297, Sep. 1995.
[33] F. Yasmin, R. Bemthuis, M. Elhagaly, F. Wijnhoven and F. Bukhsh, "A Process Mining Starting Guideline for Process Analysts and Process Owners: A Practical Process Analytics Guide using ProM," *DSI technical report series*, Jul. 2020.
[34] A. Adriansyah, "Replay a log on petri net for performance/ conformance plug-in," Technische Universiteit Eindhoven, 2012.
[35] F. Mannhardt, M Leoni and H Reijers, "The Multi-perspective Process Explorer," *BPM (Demos)*, vol. 1418, pp. 130-134, Aug. 2015.
[36] D. Chen, X. Liu, Y. Zhou, X. Yang, L. Lu and W. Xin, "Grid search as applied to the determination of Mark-Houwink parameters," *J. Appl. Polym. Sci.*, vol. 76, no. 4, pp. 481-487, 2015.
[37] J. Myerson, L. Green and M. Warusawitharana, "Area under the curve as a measure of discounting," *J. Exp. Anal. Behav.*, vol. 76, no. 2, pp. 235-243, Oct. 2001.
[38] L. Zheng, G Liu, C. Yan, C Jiang and M. Li, "Improved TrAdaBoost and its Application to Transaction Fraud Detection," *IEEE Trans. Computat. Social Syst*., vol. 7, no. 5, pp. 1304-1316, Jul. 2020.
[39] J. Cui, C Yan and C Wang, "ReMEMBeR: Ranking Metric Embedding-Based Multicontextual Behavior Profiling for Online Banking Fraud Detection," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 3, pp. 643 - 654, Aug. 2021.
[40] 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 Trans. Computat. Social Syst.*, early access, doi: 10.1109/TCSS.2022.3158318.
[41] Q. Yang, C. Wang, C. Wang, H, Teng and C Jiang, "Fundamental Limits of Data Utility: A Case Study for Data-Driven Identity Authentication," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 2, pp. 398-409, Aug. 2021.
[42] J. Liang, Y. Tang, R. Hare, B. Wu and F. Wang, "A Learning-Embedded Attributed Petri Net to Optimize Student Learning in a Serious Game," *IEEE Trans. Computat. Social Syst.*, early access, doi: 10.1109/TCSS.2021.3132355.
[43] L. He, G. Liu and M. Zhou, "Petri-Net-Based Model Checking for Privacy-Critical Multiagent Systems," *IEEE Trans. Computat. Social Syst.*, early access, doi: 10.1109/TCSS.2022.3164052.
[44] F Zhao, D. Xiang, G Liu and C Jiang, "A New Method for Measuring the Behavioral Consistency Degree of WF-Net Systems," *IEEE Trans. Computat. Social Syst.*, vol. 9, no. 2, pp. 480--493, Sep. 2022.
[45] G.J. Liu, *Petri Nets: Theoretical Models and Analysis Methods for Concurrent Systems.* Singapore, Singapore, Springer, Nov. 2022, pp. 123-165.

WangYang Yu received the Ph.D. degree from Tongji University, Shanghai, China, in 2014. He is currently an Associate Professor with the School of Computer Science, Shaanxi Normal University, Xi'an, China. His research interests include the theory of Petri nets, formal methods in software engineering, and artificial intelligence.

YaDi Wang is a postgraduate student with the School of Computer Science, Shaanxi Normal University, Xi'an, China. Her research interests include the theory of Petri nets, process mining, online transaction systems, formal methods in software engineering, and artificial intelligence.



Lu Liu is the Head of School of Computing and Mathematical Sciences at the University of Leicester, UK. Professor Liu received his PhD degree from Surrey Space Centre at the University of Surrey, UK. His research interests are in the areas of data analytics, service computing, sustainable computing and the Internet of Things. He has over 250 scientific publications in reputable journals, academic books and international conferences. Professor Liu has secured many research projects which are supported by research councils, BIS, Innovate UK, British Council and leading industries. He received the Vice-Chancellor's Award for Excellence in Doctoral Supervision in 2018, BCL Faculty Research Award in 2012 and the Promising Researcher Award in 2011. He has been the recipient of 7 Best Paper Awards from international conferences and was invited to deliver 8 keynote speeches at international conferences. Professor Liu is a Fellow of BCS (British Computer Society). He is currently serving as an Associate Editor for Peer-to-Peer Networking and Application (PPNA) and Big Data Mining and Analytics (BDMA). He has chaired over 20 international conferences in the areas of Data Science AI Cloud Computing and the Internet of Things.

YiSheng An received the M.S. and Ph.D. degrees in systems engineering from Xi'an Jiaotong University, Xi'an, China, in 2001 and 2007, respectively. He is an IEEE Member, and a Professor with the Department of Computer Science and Engineering, School of Information Engineering, Chang'an University, Xi'an. His research interests include Internet of Vehicles, intelligent transportation systems and distributed information systems.

Bo Yuan received the BEng and PhD degree in computer science from the Tongji University, Shanghai, China in 2011 and 2017, respectively. He is currently a Lecturer in Computer Science with the School of Computing and Mathematical Sciences, University of Leicester, UK. His research interests include Distributed Networks, Artificial Intelligence, Internet of Things, Federated Learning, and Edge Computing.

John Panneerselvam is a Lecturer in Informatics at the University of Leicester, UK. John received his PhD in Computing from the University of Derby in 2018 and an MSc in advanced computer networks in 2013. He is an active member of IEEE and British Computer Society, and a HEA fellow. His research interests include cloud computing, fog computing, Internet of Things, big data analytics, bioinformatics, and P2P computing.