Transaction Fraud Detection Based on Total Order Relation and Behavior Diversity

Lutao Zheng, Guanjun Liu, *Member, IEEE*, Chungang Yan, and Changjun Jiang

Abstract—**With the popularization of online shopping, transaction fraud is growing seriously. Therefore, the study on fraud detection is interesting and significant. An important way of detecting fraud is to extract the behavior profiles (BPs) of users based on their historical transaction records, and then to verify if an incoming transaction is a fraud or not in view of their BPs. Markov chain models are popular to represent BPs of users, which is effective for those users whose transaction behaviors are stable relatively. However, with the development and popularization of online shopping, it is more convenient for users to consume via the Internet, which diversifies the transaction behaviors of users. Therefore, Markov chain models are unsuitable for the representation of these behaviors. In this paper, we propose logical graph of BP (LGBP) which is a total order-based model to represent the logical relation of attributes of transaction records. Based on LGBP and users' transaction records, we can compute a path-based transition probability from an attribute to another one. At the same time, we define an information entropy-based diversity coefficient in order to characterize the diversity of transaction behaviors of a user. In addition, we define a state transition probability matrix to capture temporal features of transactions of a user. Consequently, we can construct a BP for each user and then use it to verify if an incoming transaction is a fraud or not. Our experiments over a real data set illustrate that our method is better than three state-of-the-art ones.**

Index Terms—**Behavior profile (BP), e-commerce security, fraud detection, online transaction.**

I. Introduction

THE volume of the electronic transaction has rised significantly in recent years due to the popularization of online shopping (e.g., Amazon, eBay, and Alibaba). The global e-commerce market is predicted that it will be worth a staggering US$ 24 trillion by 2019 [5]. Credit cards are widely used in online shopping, and card-not-present transactions [16] in credit card operations becomes more and more popular since web payment gateways (e.g., PayPal and AliPay) become popular. However, there has been a simultaneous growth of transaction fraud [36] which results in a dramatic impact on users. A survey of over 160 companies reveals that the number of online frauds is 12 times higher than that of the offline frauds [25], and the losses can increase yearly at double-digit rates by 2020 [14].

A physical card is not required in the scenario of online shopping and only the information of the card is enough for a transaction. Therefore, it is much easier for a fraudster to make a fraud. There are many ways by which fraudsters can illegally obtain the card information of a user: phishing (cloned websites) [3], [28], pseudobase station [39], trojan virus [15], collision attack [17], malicious insider [6], and so on. Therefore, it is very interesting and significant to study the methods of fraud detection.

Currently, there are two kinds of methods of fraud detection [19]: misuse detection and anomaly detection. The former is to collect a large database of fraudulent signatures and uses it as a reference to detect an incoming transaction. This kind of approach usually has to know the previous cases of fraud in order to obtain the different fraud patterns. Various supervised learning methods like neural networks, decision trees, logistic regression, and support vector machine are often used to obtain the fraud patterns [8], [10], [12], [27], [31], [40]. They are efficient for detecting those fraud cases that belong to the existing patterns [40]. However, they are unsuitable for the fraud transactions that were not recognized earlier. In addition, the individual behavior characteristics of each user are ignored in these methods.

For the anomaly detection methods [20], [24], [25], [30], [32], [33], [37], they usually extract a profile of normal transaction pattern for each user based on her/his legal transaction records, and then compute an acceptance degree for every incoming transaction based on the profile. The main idea of behavior profile (BP) is that different users can have different personalized behaviors due to their different identities, different incomes, different motivations, and so on. Adams *et al.* [4] also point out that different users have different transaction behavior patterns. Therefore, the anomaly detection methods are feasible.

At present, Markov chains and their extensions are often used as the models of personalized transaction BP [7], [13], [23], [26], [33], [35]. The main idea of Markov chain model of fraud detection is that some attribute values of transaction records (e.g., transaction amount and category of goods) are

Manuscript received August 4, 2017; revised May 7, 2018; accepted July 10, 2018. Date of publication August 7, 2018; date of current version September 11, 2018. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1001804, in part by Shanghai Science and Technology Innovation Action Plan Project under Grant 16511100900, and in part by the National Natural Science Foundation of China under Grant 61572360. *(Corresponding authors: Guanjun Liu; Changjun Jiang.)*

The authors are with the Key Laboratory of the Ministry of Education for Embedded System and Service Computing, Department of Computer Science, Tongji University, Shanghai 201804, China, and also with the Shanghai Electronic Transactions and Information Service Collaborative Innovation Center, Tongji University, Shanghai 201804, China (e-mail: zhenglutao103@163.com; liuguanjun@tongji.edu.cn; yanchungang@tongji.edu.cn; cjjiang@tongji.edu.cn).

Digital Object Identifier 10.1109/TCSS.2018.2856910

as the nodes of a model and the transition probabilities among those nodes are used to quantify the transaction behavior features. Markov chain models are good at describing their BPs for those users whose transaction behaviors are stable, and our experiences also prove this. However, with the development and popularization of online shopping, the transaction behaviors of a user vary often and then her/his BP should be able to characterize transaction diversity. Therefore, Markov chain models are not too suitable for those users. In this paper, we propose a new model to represent a user's BP under considering behavior diversity, and then present a fraud detection method based on this new model. The main contributions of this paper are summarized as follows.

1) First, we totally order the attributes of transaction records, and then classify the values of every attribute. Based on them, we construct a logical graph of BP (LGBP) which abstracts and covers all different transaction records.
2) Based on LGBP, we define the path-based transition probability and diversity coefficient to characterize users' transaction behaviors and diversity. We also define a state transition probability matrix to capture temporal features of transactions, and then construct a BP for each user.
3) A BP-based fraud detection method is proposed to determine the legality of an incoming transaction, and it considers the concept drift problem [2].
4) Experiments are conducted to evaluate the performance of our method (OM) compared with three anomaly detection methods.

The rest of the paper is organized as follows. The BP is introduced in Section II. Fraud detection method is proposed in Section III. Experiments and performance comparison are illustrated in Section IV. Finally, we conclude our work in Section V.

II. Behavior Profile

This section first introduces the concepts of transaction record and transaction log that are used in the construction of BP.

Definition 1 (Transaction Record): A transaction record r consists of m attribute values, i.e., $r = \{a_1, a_2, \ldots, a_m | a_1 \in A_1, a_2 \in A_2, \ldots, a_m \in A_m\}$ where $A_i = \{a_1^i, a_2^i, \ldots, a_{n_i}^i\}$ is the set of values of the ith attribute and $n_i = |A_i|$.

Given a user u, her/his transaction log is a set of her/his transaction records in a period of time and denoted as $L_u = \{r_1^u, r_2^u, \ldots, r_{n^u}^u\}$ in which $n^u = |L_u|$.

For example, Table I lists six transaction records of a user who bought goods via the online shopping website TaoBao.[1] Transaction record r_1^u means that the user bought a good of daily supply (DS) at Shanghai Jiading (SJ) at night (NI), its price is in (0, 200] Chinese Yuan, and it is shipped to Anhui Xuancheng (AX).

Note that we preprocess some information in the original records. For example, the goods are classified and transaction time is divided into four segments. Any two original records are different since their transaction time are different, but some records in L_u are possibly equal because their transaction times are put into the same segment or their goods belong to the same category. These equal records are all kept in L_u in order to characterize the user's behavior. To represent some equations conveniently, we denote R_u as the set of all different records in L_u. In fact, R_u is a set and L_u is a multiset.

In Table I, every transaction record has five attributes: *transaction_time*, *transaction_location*, *category_of_good*, *amount*, and *shipping_address*, where *transaction_time* = {Early Morning: [0, 6], Morning: (6, 12], Afternoon: (12, 18], Night: (18, 24]}, and *amount* = {(0, 200], (200, 500], (500, 1000], (1000, ∞)}.

Obviously, some attributes in each record are dependent on the related operations (events) executed in the system. Aalst *et al.* [1] point out that these events/operations can be totally ordered. For example, $transaction\_location$ can be previous to $category\_of\_good$ because a user can select goods only after s/he logins. Note that transaction time and location are recorded once s/he logins. Then, the amount of goods is obtained. At last, the user submits the shipping address. In addition, some attributes are related to a user's behavior habit. For example, it is often the morning or afternoon when a user logins at the office, while it is often the evening when s/he logins at home. Therefore, we assume that $transaction\_time$ is previous to $transaction\_location$. Without loss of generality, we let $A_1 \prec A_2 \prec \cdots \prec A_m$ in this paper. For example, the five attributes in Table I are totally ordered as follows: *transaction_time* ≺ *transaction_location* ≺ *category_of_good* ≺ *amount* ≺ *shipping_address*.

Based on the total order relation and the transaction log of a user, we can construct a logic graph of BP (LGBP) for the user, which represents the dependent relations of all attribute values of this user's records and covers all transaction records. First, we abstract all attribute values occurring in the transaction records of user u as follows:

$$\begin{aligned} A_1^u &= \{a \in A_1 | \exists r \in R_u : a \in r\} \\ A_2^u &= \{a \in A_2 | \exists r \in R_u : a \in r\} \\ &\ldots \\ A_m^u &= \{a \in A_m | \exists r \in R_u : a \in r\}. \end{aligned}$$

Obviously, $A_1^u \subseteq A_1$, $A_2^u \subseteq A_2, \ldots,$ and $A_m^u \subseteq A_m$. Without loss of generality, we denote $A_i^u = \{a_1^i, a_2^i, \ldots, a_{n_i^u}^i\}$ in which $n_i^u = |A_i^u|$ for each $i \in \{1, 2, \ldots, m\}$.

Definition 2 (LGBP): Let $L_u = \{r_1^u, r_2^u, \ldots, r_{n^u}^u\}$ be the transaction log of user u. The LGBP of u is a directed acyclic graph $G_u = (V_u, E_u)$, where:

1) $V_u = \{a_s, a_e\} \cup A_1^u \cup A_2^u \cup \cdots \cup A_m^u$ in which a_s and a_e are the two special nodes that represent the start and end of a transaction;
2) $\forall a \in A_1^u, (v_s, a) \in E_u$;
3) $\forall a \in A_m^u, (a, v_e) \in E_u$;
4) $\forall i \in \{1, 2, \ldots, m-1\}$, $\forall a \in A_i^u$, $\forall a' \in A_{i+1}^u$: $(a, a') \in E_u$ if and only if $\exists r \in R_u$: $a \in r \wedge a' \in r$.

[1] NI: night, MO: morning, AF: afternoon, SJ and AX: the abbreviations of two places' names, DS: daily supply, EP: electronic product, SS: school supply.

TABLE I
EXAMPLE OF TRANSACTION LOG

Transaction records	Transaction attributes				
	$transaction\_time$	$transaction\_location$	$category\_of\_good$	$amount$	$shipping\_address$
r_1^u	NI	SJ	DS	$(0, 200]$	AX
r_2^u	MO	SJ	SS	$(0, 200]$	SJ
r_3^u	AF	AX	DS	$(0, 200]$	AX
r_4^u	MO	SJ	EP	$(500, 1000]$	SJ
r_5^u	MO	SJ	SS	$(0, 200]$	SJ
r_6^u	NI	SJ	DS	$(0, 200]$	SJ

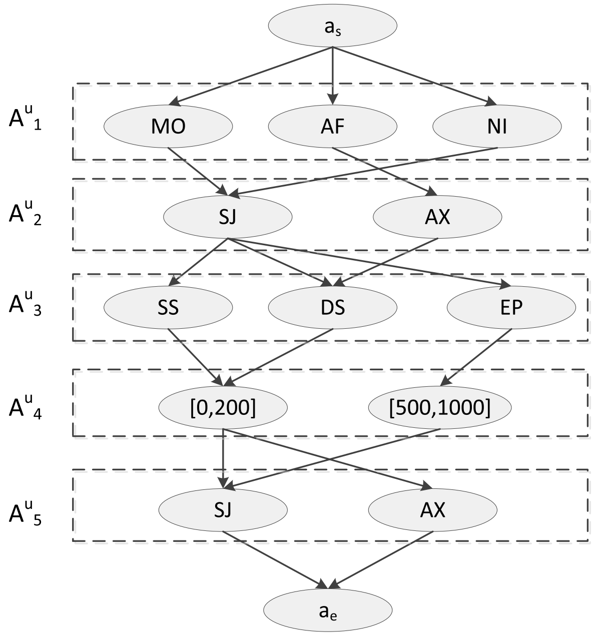


Fig. 1. LGBP of a user whose transaction log is listed in Table I.

Fig. 1 shows the LGBP of a user whose transaction log is listed in Table I.

Definition 3 (Prepaths): Let $G_u = (V_u, E_u)$ be the LGBP of user u. $\forall v \in V_u$, prepaths(v) is the set of all directed paths from node a_s to node v in G_u.

Definition 4 (Postnodes): Let $G_u = (V_u, E_u)$ be the LGBP of u. $\forall v \in V_u$, postnodes(v) is the set of nodes that are directly reached from v in G_u.

For example, there are five directed paths from node a_s to (0, 200] in Fig. 1

$$\sigma_1 = a_s \cdot \text{MO} \cdot \text{SJ} \cdot \text{SS} \cdot (0, 200]$$
$$\sigma_2 = a_s \cdot \text{MO} \cdot \text{SJ} \cdot \text{DS} \cdot (0, 200]$$
$$\sigma_3 = a_s \cdot \text{AF} \cdot \text{AX} \cdot \text{DS} \cdot (0, 200]$$
$$\sigma_4 = a_s \cdot \text{NI} \cdot \text{SJ} \cdot \text{SS} \cdot (0, 200]$$
$$\sigma_5 = a_s \cdot \text{NI} \cdot \text{SJ} \cdot \text{DS} \cdot (0, 200].$$

Nodes SJ and AX in A_5^u are the postnodes of *(0,200]* in Fig. 1, i.e., postnodes((0,200]) = {SJ, AX}.

Definition 5 (Path-Based Probability Transition Matrix): Let $G_u = (V_u, E_u)$ be the LGBP of user u. $\forall v \in V_u$, M_v is a $|prepaths(v)| \times |\text{postnodes}(v)|$ matrix where $\forall \sigma \in prepaths(v)$, $\forall v' \in \text{postnodes}(v)$

$$M_v(\sigma, v') = P(v \to v'|\sigma) \tag{1}$$

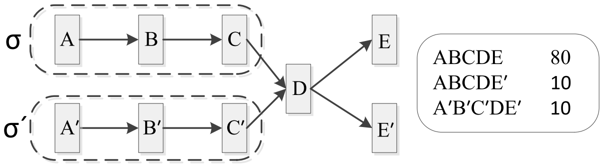


Fig. 2. Example to illustrate the transition probability between two transaction features.

which is the transition probability from v to v' under the condition that v is reached via σ.

For example, node *(0, 200]* in Fig. 1 has five directed paths from a_s: $\sigma_1 - \sigma_5$, and has two postnodes: SJ and AX. Therefore, we have

$$M_{(0,200]} = \begin{pmatrix} P((0,200] \to \text{SJ}|\sigma_1) & P((0,200] \to \text{AX}|\sigma_1) \\ P((0,200] \to \text{SJ}|\sigma_2) & P((0,200] \to \text{AX}|\sigma_2) \\ P((0,200] \to \text{SJ}|\sigma_3) & P((0,200] \to \text{AX}|\sigma_3) \\ P((0,200] \to \text{SJ}|\sigma_4) & P((0,200] \to \text{AX}|\sigma_4) \\ P((0,200] \to \text{SJ}|\sigma_5) & P((0,200] \to \text{AX}|\sigma_5) \end{pmatrix}.$$

Here, we use the path-based transition probability which is different from the transition probability of the method in [33]. OM can characterize the user's transaction behavior more precisely. For example, as shown in Fig. 2, there are 100 transaction records in which $ABCDE$ takes place 80 times, $ABCDE'$ takes place 10 times, and $A'B'C'DE'$ take places 10 times. If we use the method in [33] to compute the transition probabilities from D to E and E', they are 80% and 20%, respectively. Other transition probabilities (e.g., from A to B) are all 100%. If we use OM, the transition probabilities from D to E and E' will consider two different prepaths ($\sigma = ABC$ and $\sigma' = A'B'C'$), and then we will obtain four transition probabilities: $P(D \to E|\sigma) = 80\%$, $P(D \to E'|\sigma) = 10\%$, $P(D \to E|\sigma') = \omega_u$ (ω_u is a diversity coefficient that will be introduced later), and $P(D \to E'|\sigma') = 10\%$. Other path-based transition probabilities are all 100%. For $A'B'C'DE$, its occurrence probability is $P(A' \to B') \times P(B' \to C') \times P(C' \to D) \times P(D \to E) = 80\%$ according to the method in [33]. However, this transaction has never taken place, and thus the occurrence probability 80% is not reasonable. Its occurrence probability computed by OM is much less than 80% and greater than 0, which is more reasonable. In fact, this erroneous conclusion based on (Hidden Markov Model [33]) can be avoided by defining some proper internal states to compute the transition probability. However, this possibly results in a large number of internal

TABLE II

ILLUSTRATION OF DIFFERENT VALUES OF DIVERSITY UNDER DIFFERENT DISTRIBUTIONS OF RECORDS

	r_1	r_2	r_3	r_4	r_5	r_6	ω_u
Case1	1	1	1	1	1	1	1
Case2	1	3	1	1	0	0	0.35
Case3	3	3	0	0	0	0	0.2
Case4	5	1	0	0	0	0	0.13
Case5	6	0	0	0	0	0	0

states. Fortunately, the path-based method defined in this paper can avoid these internal states.

In what follows, we propose a method to calculate the path-based transition probability matrix for each node. We first introduce a diversity coefficient ω_u to reflect the diversity of transaction behaviors of a user u. In other words, the greater the value of ω_u, the higher the probability of conducting a transaction which never took place in the historical records of the user. Referring to the information entropy [9], we use the following equation to represent the diversity:

$$\omega_u = -\sum_{i \in R_u} P(r) \times \log_\kappa P(r) \tag{2}$$

where $P(r)$ is the probability of record r occurring in L_u and is calculated by the frequency of r in L_u, U is the set of all users, and

$$\kappa = \max_{u \in U} \{|R_u|\}.$$

Table II illustrates five different values of ω_u if the user u has six transaction records, and we consider five different cases of these records. For example, Case 2 means that r_2 occurs three times in the six records, and r_1, r_3, and r_4 occur once, respectively. Hence, $\omega_u = 0.35$ for Case 2 where we let $\kappa = 32$. Obviously, the value of ω_u becomes greater if the diversity of transaction records of a user is more abundant.

Given a directed path σ in the LGBP of user u, we use the following equation to represent the frequency of σ occurring in all transaction records. Note that σ occurs in a record r if and only if all nodes (except for a_s) in σ are in r:

$$f(\sigma) = |\{r \in L_u | \forall v \in \#(\sigma)/\{a_s\} : v \in r\}|. \tag{3}$$

In (3), $\#(\sigma)$ is the set of all nodes in σ. Considering the diversity of transaction behaviors of user u, we use the following equation to calculate the transition probability from v to v' under the condition that v is reached from a_s via σ:

$$P(v \to v'|\sigma) = \begin{cases} (1-\omega_u) \times \dfrac{f(\sigma v')}{f(\sigma)} & f(\sigma) \neq 0 \\ 0 & \mathrm{f}(\sigma) = 0. \end{cases} \tag{4}$$

If (4) has not coefficient $1 - \omega_u$, then we have that

$$\begin{aligned} \sum_{v' \in \text{postnodes}(v)} P(v \to v'|\sigma) &= \sum_{v' \in \text{postnodes}(v)} \frac{f(\sigma v')}{f(\sigma)} \\ &= \frac{\sum_{v' \in \text{postnodes}(v)} f(\sigma v')}{f(\sigma)} \\ &= \frac{f(\sigma)}{f(\sigma)} = 1. \end{aligned}$$

This means that in the future, the user cannot conduct a new transaction that never took place in her/his historical records, which contradicts the fact. Therefore, we use a diversity coefficient to represent the probability that a user conducts a new transaction.

$f(\sigma) = 0$ means that there is no record r in L_u such that

$$\forall v'' \in \#(\sigma)/\{a_s\} : v'' \in r.$$

Hence, the transition probability from v to v' under σ should be equal to zero, i.e., $P(v \to v'|\sigma) = 0$ if $f(\sigma) = 0$ in (4).

For example, in the five paths $\sigma_1 - \sigma_5$ from node a_s to node (0, 200] in Fig. 1, σ_1 corresponds to r_2^u and r_5^u , σ_3 corresponds to r_3^u, σ_5 corresponds to r_1^u and r_6^u, but σ_2 and σ_4 do not correspond to any record. Therefore, we can obtain the path-based transition probability matrix of node (0, 200] as follows:

$$M_{(0,200]} = \begin{array}{c} \\ \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \end{array} \!\! \begin{array}{c} \begin{array}{cc} \text{SJ} \quad\quad & \text{AX} \end{array} \\ \begin{pmatrix} 1-\omega_u & 0 \\ 0 & 0 \\ 0 & 1-\omega_u \\ 0 & 0 \\ \dfrac{1-\omega_u}{2} & \dfrac{1-\omega_u}{2} \end{pmatrix} \end{array}.$$

$P((0,200] \to \text{SJ}|\sigma_1) = 1 - \omega_u$ due to $f(\sigma_1) = 2$ and $f(\sigma_1\text{SJ}) = 2$. $P((0,200] \to \text{AX}|\sigma_1) = 0$ due to $f(\sigma_1) = 2$ and $f(\sigma_1\text{AX}) = 0$. $P((0,200] \to \text{SJ}|\sigma_4) = P((0,200] \to \text{AX}|\sigma_4) = 0$ due to $f(\sigma_4) = 0$. One can easily understand others.

Next, we define state transition probability matrix to capture the temporal features of transactions of a user. The temporal features can be depicted by many methods, e.g., window sliding-based method [18]. Srivastava *et al.* [33] point out that a user makes purchases depending on his/her needs for procuring different categories of goods over a period of time. Therefore, they treat categories of goods as the states of their Markov chain model in order to capture the temporal features. In this paper, we use their method to construct the transition probabilities of good categories as the temporal features of transactions of a user.

Definition 6 (State Transition Probability Matrix): Let $\mathcal{C} = \{C_1, C_2, \ldots, C_N\}$ be the set of categories of goods. The good category in the transaction at time t is denoted by q_t. The state transition probability matrix with respect to user u is denoted as $\mathcal{T}_u = [\tau_{ij}]$, where

$$\tau_{ij} = P(q_{t+1} = C_j | q_t = C_i), \quad 1 \leq i, \ j \leq N.$$

Specifically

$$P(q_{t+1} = C_j | q_t = C_i) = \frac{f(C_i \to C_j)}{f(C_i)}$$

where

$$f(C_i) = \left|\left\{r \in L_u / \left\{r_{n^u}^u\right\} \middle| C_i \in r\right\}\right|$$

and

$$f(C_i \to C_j) = |\{\{r, r'\} \subseteq L_u | C_i \in r \wedge C_j \in r'\}|.$$

Note that r and r' are the two consecutive transactions in L_u such that r occurs before r'. Therefore, we only compute the transition probability of two good categories in two consecutive transactions. For example, in Table I, there are three categories of goods: DS, SS, and EP. The transition probability of DS and SS that are in two consecutive transactions r_1^u and r_2^u can be calculated as follows:

$$P(q_2 = \text{SS}|q_1 = \text{DS}) = \frac{1}{2} = 0.5$$

and then, we can obtain the state transition probability matrix of Table I as follows:

$$\mathcal{T} = \begin{array}{c} \\ \text{DS} \\ \text{SS} \\ \text{EP} \end{array} \begin{array}{c} \begin{array}{ccc} \text{DS} & \text{SS} & \text{EP} \end{array} \\ \begin{pmatrix} 0 & 0.5 & 0.5 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \end{array}.$$

Obviously, $\sum_{j=1}^{N} \tau_{ij} = 1$ for each $i \in \{1, \ldots, N\}$. In what follows, we use $C(r)$ to represent the category of the good occurring in transaction r.

Definition 7 (Behavior Profile): Let $L_u = \{r_1^u, r_2^u, \ldots, r_{n^u}^u\}$ be the transaction log of user u. $\text{BP}_u = (V_u, E_u, \mathcal{M}_u, \mathcal{T}_u, \omega_u)$ is the BP of u, where:

1) $G_u = (V_u, E_u)$ is the LGBP of u;
2) $\mathcal{M}_u = \{M_\upsilon | \upsilon \in V_u\}$ is the set of path-based transition probability matrices of all nodes in G_u;
3) $\mathcal{T}_u$ is the state transition probability matrix w.r.t. u, and
4) ω_u is the diversity coefficient of u.

For each user, we can construct a BP based on her/his transaction log. In what follows, we propose a method to detect if a transaction record is acceptable by a BP.

III. Detection of Fraud

We first propose an algorithm to compute the recognization degree of a transaction record by a given BP. Let $\text{BP}_u = (V_u, E_u, \mathcal{M}_u, \mathcal{T}_u, \omega_u)$ be the BP of user u and $r = \{a_1, a_2, \ldots, a_m\}$ be an arbitrary record where $\forall i \in \{1, 2, \ldots, m\}$: $a_i \in A_i$. We denote $\sigma_r = a_0 \cdot a_1 \cdots a_m \cdot a_{m+1}$ as the virtual path corresponding to r, and denote $\sigma_r^i = a_0 \cdot a_1 \cdots a_{i-1}$ as the prefix of σ_r with length i, where $a_0 = a_s$ and $a_{m+1} = a_e$.

If $r \in L_u$, we know that $f(\sigma_r^i) \neq 0$ and thus $P(a_{i-1} \to a_i|\sigma_r^i) \neq 0$, $\forall i \in \{1, 2, \ldots, m\}$. Hence, we can use the following equation to calculate the recognization degree of r by BP_u:

$$\beta(r, \text{BP}_u) = \prod_{i=1}^{m} P\big(a_{i-1} \to a_i|\sigma_r^i\big) = \prod_{i=1}^{m} M_{a_{i-1}}\big(\sigma_r^i, a_i\big). \tag{5}$$

Obviously, $0 < \beta(r, \text{BP}_u) \leq 1$. Based on (3) and (4), we can derive

$$\sum_{r \in R_u} \beta(r, \text{BP}_u) = (1 - \omega_u)^m.$$

If there is no coefficient $1 - \omega_u$ in (4), then

$$\sum_{r \in R_u} \beta(r, \text{BP}_u) = 1.$$

Algorithm 1 Calculation of Recognization Degree

```
Input: A user's BP $BP_u = (V_u, E_u, \mathcal{M}_u, \mathcal{T}_u, \omega_u)$ and
       a transaction record $r = \{a_1, a_2, \ldots, a_m\}$ where
       $\forall i \in \{1, 2, \ldots, m\}$: $a_i \in A_i$;
Output: The recognization degree $\beta$ of $r$ by $BP_u$;
$a_0 := a_s$;
$\sigma := a_0$;
$\beta := 1$;
for (i:=1; $i \leq m$; i++) do
    if $M_{a_{i-1}}(\sigma, a_i) \neq 0$ then
        $\beta := \beta \times M_{a_{i-1}}(\sigma, a_i)$;
        $\sigma := \sigma \cdot a_i$;
    else
        $\beta := \beta \times \omega_u$;
        select node $\upsilon_{\max}$ that satisfies $M_{a_{i-1}}(\sigma, \upsilon_{\max}) =$
        $\max_{\upsilon \in postnodes(a_{i-1})}\{M_{a_{i-1}}(\sigma, \upsilon)\}$;
        $\sigma := \sigma \cdot \upsilon_{\max}$;
    end
end
```

The above-mentioned conclusion implies that the recognization degree of a transaction represents the probability of the transaction in the history (under considering the transaction diversity).

If $r \notin L_u$, then we know that $f(\sigma_r) = 0$. In fact, there are two cases leading to $f(\sigma_r) = 0$. One is that $\forall a \in r$: $a \in V_u$, but $\exists i \in \{1, 2, \ldots, m\}$: $f(\sigma_r^i) \neq 0 \wedge f(\sigma_r^{i+1}) = 0$. Another one is that $\exists a \in r$: $a \notin V_u$. For the latter, we still have that $\exists i \in \{1, 2, \ldots, m\}$: $f(\sigma_r^i) \neq 0 \wedge f(\sigma_r^{i+1}) = 0$. The two cases both result in $P(a_{i-1} \to a_i|\sigma_r^i) = 0$, where $i \in \{1, 2, \ldots, m\}$. Hence, if we use (5) to calculate the recognization degree of r by BP_u, then $\beta(r, \text{BP}_u) = 0$. However, if we use (5) to measure the degree that a transaction which never took place in the historic records is similar to the historic behavior, we do not hope that its recognization degree is 0. Therefore, $P(a_{i-1} \to a_i|\sigma_r^i)$ is assigned by ω_u instead of 0 for the case of $r \notin L_u$. For the next calculation, we should update σ_r^{i+1} since $f(\sigma_r^{i+1}) = 0$. $\sigma_r^{i+1} = \sigma_r^i a_i$ is replaced by $\sigma_r^{i+1} = \sigma_r^i \upsilon_{\max}$ such that

$$M_{a_{i-1}}\big(\sigma_r^i, \upsilon_{\max}\big) = \max_{\upsilon \in \text{postnodes}(a_{i-1})} \big\{M_{a_{i-1}}\big(\sigma_r^i, \upsilon\big)\big\}. \tag{6}$$

The above-mentioned equation guarantees $f(\sigma_r^{i+1}) = f(\sigma_r^i \upsilon_{\max}) \neq 0$.

Algorithm 1 can clearly describe the process of calculating the recognization degree of a transaction.

For the BP shown in Table I and Fig. 1, we consider the following three examples whose recognization degrees are computed by Algorithm 1.

Because transaction record r_1^u = {NI, SJ, DS, (0, 200], AX} in Table I belongs to the transaction log, we can easily calculate its recognization degree as follows:

$$\begin{aligned}\beta(r_1^u, \mathrm{BP}_u) &= M_{a_s}(a_s, \mathrm{NI}) \times M_{\mathrm{NI}}(a_s \cdot \mathrm{NI}, \mathrm{SJ})\\ &\quad \times M_{\mathrm{SJ}}(a_s \cdot \mathrm{NI} \cdot \mathrm{SJ}, \mathrm{DS})\\ &\quad \times M_{\mathrm{DS}}(a_s \cdot \mathrm{NI} \cdot \mathrm{SJ} \cdot \mathrm{DS}, (0.200])\\ &\quad \times M_{(0,200]}(a_s \cdot \mathrm{NI} \cdot \mathrm{SJ} \cdot \mathrm{DS} \cdot (0.200], \mathrm{AX})\\ &= \frac{1-\omega_u}{3} \times (1-\omega_u) \times (1-\omega_u) \times (1-\omega_u)\\ &\quad \times \frac{1-\omega_u}{2} = \frac{(1-\omega_u)^5}{6}.\end{aligned}$$

Transaction $r' = \{\mathrm{NI}, \mathrm{SJ}, \mathrm{SS}, (0, 200], \mathrm{SJ}\}$ is not in the transaction log but there is path $a_s \cdot \mathrm{NI} \cdot \mathrm{SJ} \cdot \mathrm{SS} \cdot (0, 200] \cdot \mathrm{SJ} \cdot a_e$ in Fig. 1 corresponding to it. Due to $P(\mathrm{SJ} \rightarrow \mathrm{SS} | a_s \cdot \mathrm{NI} \cdot \mathrm{SJ}) = 0$, we should specially consider node SJ for this transaction. The process of computing its recognization degree as follows:

$$\begin{aligned}\beta(r', \mathrm{BP}_u) &= M_{a_e}(a_e, \mathrm{NI}) \times M_{\mathrm{NI}}(a_e \cdot \mathrm{NI}, \mathrm{SJ}) \times \omega_u\\ &\quad \times M_{\mathrm{DS}}(a_e \cdot \mathrm{NI} \cdot \mathrm{SJ} \cdot \mathrm{DS}, (0.200])\\ &\quad \times M_{(0,200]}(a_e \cdot \mathrm{NI} \cdot \mathrm{SJ} \cdot \mathrm{DS} \cdot (0, 200], \mathrm{SJ})\\ &= \frac{1-\omega_u}{3} \times (1-\omega_u) \times \omega_u \times (1-\omega_u) \times \frac{1-\omega_u}{2}\\ &= \frac{\omega_u(1-\omega_u)^4}{6}.\end{aligned}$$

In the above-mentioned computation process ω_u is used because $M_{\mathrm{SJ}}(a_e \cdot \mathrm{NI} \cdot \mathrm{SJ}, \mathrm{SS}) = 0$. In the next step, path $a_s \cdot \mathrm{NI} \cdot \mathrm{SJ} \cdot \mathrm{DS}$ replaces path $a_s \cdot \mathrm{NI} \cdot \mathrm{SJ} \cdot \mathrm{SS}$ due to $M_{\mathrm{SJ}}(a_e \cdot \mathrm{NI} \cdot \mathrm{SJ}, \mathrm{DS}) > M_{\mathrm{SJ}}(a_e \cdot \mathrm{NI} \cdot \mathrm{SJ}, \mathrm{EP})$.

We consider transaction $r'' = \{\mathrm{EM}, \mathrm{SJ}, \mathrm{SS}, (0, 200], \mathrm{SJ}\}$ that is not in the transaction log and $\mathrm{EM} \in r''$ is not in BP_u. Therefore, we also should specially consider node EM for this transaction. The process of computing its recognization degree as follows:

$$\begin{aligned}\beta(r'', \mathrm{BP}_u) &= \omega_u \times M_{\mathrm{MO}}(a_s \cdot \mathrm{MO}, \mathrm{SJ})\\ &\quad \times M_{\mathrm{SJ}}(a_s \cdot \mathrm{MO} \cdot \mathrm{SJ}, \mathrm{SS})\\ &\quad \times M_{\mathrm{SS}}(a_s \cdot \mathrm{MO} \cdot \mathrm{SJ} \cdot \mathrm{SS}, (0.200])\\ &\quad \times M_{(0,200]}(a_s \cdot \mathrm{MO} \cdot \mathrm{SJ} \cdot \mathrm{SS} \cdot (200], \mathrm{SJ})\\ &= \omega_u \times (1-\omega_u) \times \frac{2(1-\omega_u)}{3}\\ &\quad \times (1-\omega_u) \times (1-\omega_u) = \frac{2\omega_u(1-\omega_u)^4}{3}.\end{aligned}$$

In the above-mentioned computation process, ω_u is used because node EM is not in the BP_u and thus there is no edge from a_s to EM. In the next step, path $a_s \cdot \mathrm{MO}$ replaces path $a_s \cdot \mathrm{EM}$ due to $M_{a_s}(a_s, \mathrm{MO}) > M_{a_s}(a_s, \mathrm{NI}) > M_{a_s}(a_s, \mathrm{AF})$.

Then, we compute the recognization degree, we can use the following equation to calculate the acceptance degree of r by BP_u:

$$\varphi(r, \mathrm{BP}_u) = \beta(r, \mathrm{BP}_u) \times P\left(q_{n^u+1} = C(r) \middle| q_{n^u} = C\left(r_{n^u}^u\right)\right). \quad (7)$$

Next, we give a method to decide if an incoming transaction is legal or not based on acceptance degree. We calculate the acceptance degrees of the latest k transaction records and then obtain their mean φ_k. For an incoming transaction r, we calculate its acceptance degree φ. We use the following inequation to evaluate the illegality of this transaction:

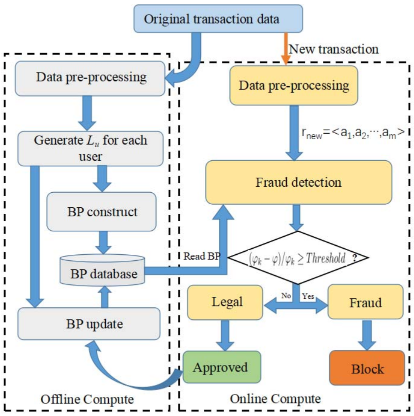


Fig. 3. Framework of yielding BP, checking legality and updating BP.

$$(\varphi_k - \varphi)/\varphi_k \geq \text{Threshold}. \quad (8)$$

If $(\varphi_k - \varphi)/\varphi_k \geq$ threshold, then this transaction is thought of as a fraud, else it is legal. Later our experiments will show that how to select k and threshold. Note that the idea of using $(\varphi_k - \varphi)/\varphi_k \geq$ threshold comes from literature [33], but (8) is different from the one in [33]. That is, (8) uses the means of acceptance degrees of the latest k records but the inequation in [33] only uses the acceptance degree of the latest one. The reason is that OMs can well cope with the problem of concept drift [2] (i.e., the sudden change of user's transaction behavior).

We can use some existing policies such as event-driven one [11] and time-driven one [22] to update a user's BP. Here, we do not introduce them in detail. Fig. 3 illustrates the framework of yielding BP, detecting fraud online, and updating BP.

IV. Performance Evaluation

In this section, we exhibit the performance of the method proposed in this paper. First, we introduce our data set and set parameters. Then, we illustrate the comparison results.

A. Dateset and Parameters

In real life, credit card data are usually unavailable for us. Although there are some public data sets about credit card fraud detection such as the one in https://www.kaggle.com/dalpozz/creditcardfraud, the values in these data sets have been modified very much, and we cannot know most of their attributes. Specially, every user has several records in these data sets. In a word, these data sets are not suitable for our experiments. Therefore, we collect some real data from 70 users. For each user, our data set contains her/his 70 transaction records of shopping on TaoBao. The way of doing experiments with self-collection data has been taken in many literatures such as the famous one of intrusion detection in [29].

TABLE III

CONFUSION MATRIX

	Labeled as fraudulent	Labeled as genuine
Actual fraudulent	True positives	False negatives
Actual negative	False positives	True negatives

We use five attributes of transaction records: *transaction_time*, *transaction_location*, *category_of_goods*, *amount*, and *shipping_address*. Note that the five attributes are shown to be effective for transaction fraud detection [18], [25], [33]. Here, a whole day (24 h is divided into four parts: 0∼6 (Early Morning, EM), 6∼12 (Morning, MO), 12∼18 (Afternoon, AF), 18∼24 (Night, NI). Categories of goods include daily supplies, school supplies, sport goods, food and others. Money amount is divided into four segments: (0, 200], (200, 500], (500, 1000], and (1000, ∞). In fact, the data used in our experiments are preprocessed based on the earlier transformations.

We randomly select 50 users. For each user, we use her/his latest N ($N = 35, 40, 45, 50$) records in her/his earliest 50 records as the training set to construct her/his BP, and we use the latest 20 records as legal transactions in her/his test set. Furthermore, from all records of the remaining 20 users, we randomly select 20 records as fraudulent transactions in her/his test set. We first introduce two performance metrics true positive rate (TPR) and false positive rate (FPR) [34].

As given in Table III, TPR is the percentage of fraudulent transactions caught by the system, and FPR is the percentage of legal transactions labeled as fraudulent (also called false alarms). It is important for a method to achieve a high TPR along with a low FPR. However, most of the methods that lead to higher TPR also increase FPR [25], [33]. Therefore, we empirically determine the design parameters of our system: N (the number of records in a training set), k, and threshold in (8), and then perform a set of experiments to choose a good combination which can result in the best performance of TPR and FPR. Due to the influence of concept drift [2], the useful information of early records may diminish as time passes by. In other words, using $N + 1$ transactions to construct BP is not much more accurate than using N ones. Here, N is set by 35, 40, 45, and 50, respectively. Consequently, we select a proper N through experiments. Because a user's transaction behavior is of a possibility of sudden change, we use the mean of acceptance degrees of the latest k transaction records as a baseline to measure the legality of the current transaction. Here, k in (8) is from 5 to 10. Similarly, threshold in (8) is set by 81%, 84%, 87%, 90%, 93%, and 96%, respectively. We need to try every combination of N, k, and threshold and then select a good one. We do a set of experiments to determine the best combination of those parameters.

TABLE IV

VALUES OF TPR AND FPR FOR DIFFERENT THRESHOLDS AND N WITH THE FIXED $k = 8$

	N=35		N=40		N=45		N=50	
T%	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
81	0.882	0.130	0.915	0.142	0.878	0.134	0.893	0.139
84	0.894	0.134	0.923	0.145	0.903	0.137	0.918	0.161
87	0.918	0.149	0.937	0.153	0.919	0.151	0.924	0.169
90	0.937	0.152	0.948	0.154	0.932	0.153	0.936	0.171
93	0.938	0.163	0.949	0.163	0.935	0.171	0.940	0.188
96	0.938	0.175	0.949	0.178	0.935	0.182	0.940	0.195

TABLE V

VALUES OF TPR AND FPR FOR DIFFERENT k AND N WITH THE FIXED THRESHOLD = 90%

	N=35		N=40		N=45		N=50	
k	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
5	0.892	0.143	0.924	0.157	0.915	0.162	0.909	0.163
6	0.912	0.149	0.938	0.159	0.924	0.164	0.913	0.167
7	0.928	0.150	0.943	0.162	0.928	0.167	0.924	0.170
8	0.938	0.152	0.948	0.154	0.932	0.153	0.936	0.171
9	0.929	0.163	0.942	0.161	0.926	0.167	0.927	0.178
10	0.932	0.174	0.943	0.165	0.931	0.169	0.928	0.187

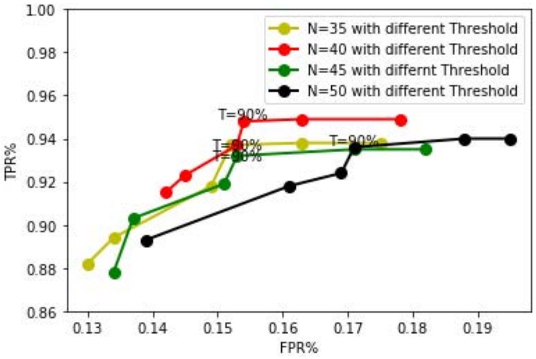


Fig. 4. TPR and FPR for different thresholds and N where $k = 8$.

Table IV shows the values of TPR/FPR for different N and threshold with the fixed $k = 8$. Similarly, Table V shows the values of TPR/FPR for different k and N with the fixed threshold = 90%, and Table VI shows the values of TPR/FPR for different thresholds and k with the fixed $N = 40$. In fact, there are $4 + 6 + 6 = 16$ tables totally and we omit them to save space. In addition, in order to make a clear observation of these results, we can present an equivalent line chart for each table, as shown in Figs. 4–6 which correspond to Tables IV–VI, respectively.

As shown in Figs. 4–6, TPR increases when FPR increases. For each line chart, the best case is such a point that is close to the top-left corner. Our results show that most of the best cases correspond to the point that $N = 40$, $k = 8$, and threshold = 90%. Therefore, the three parameters are used in the next comparison experiments.

B. Comparative Performance

We compare OM with the famous fraud detection method proposed by Srivastava *et al.* [33]. This method is called Srivastava's method (SM) in this paper, and OM is called OM. SM uses Markov chain as the model of BP and do experiments through their simulation data. Here, we use these real data to do comparison experiments. In addition, we compare OM with other two anomaly detection methods: Bayesian learning-based fraud detection [25] and self-organizing maps-based fraud detection [38]. The two methods are called phase modulation (PM) and Vladimir Zaslavsky's method (VM), respectively.

TABLE VI

VALUES OF TPR AND FPR FOR DIFFERENT THRESHOLDS AND k WITH THE FIXED $N = 40$

	k=5		k=6		k=7		k=8		k=9		k=10	
T%	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
81	0.895	0.145	0.892	0.135	0.889	0.140	0.915	0.142	0.894	0.156	0.902	0.153
84	0.923	0.170	0.914	0.165	0.913	0.164	0.923	0.145	0.917	0.157	0.913	0.156
87	0.928	0.172	0.925	0.169	0.923	0.171	0.937	0.153	0.927	0.167	0.921	0.163
90	0.936	0.165	0.943	0.170	0.940	0.178	0.948	0.154	0.936	0.175	0.930	0.165
93	0.942	0.186	0.946	0.176	0.948	0.179	0.949	0.163	0.952	0.178	0.957	0.173
96	0.967	0.203	0.960	0.180	0.958	0.183	0.949	0.178	0.953	0.187	0.958	0.183

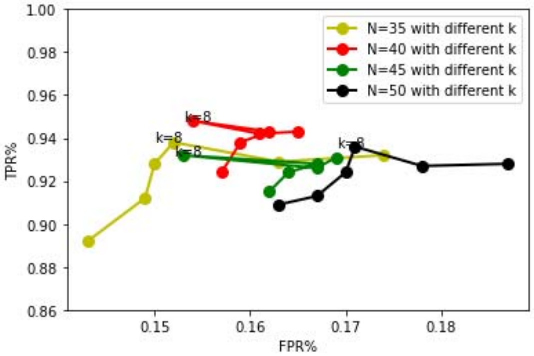


Fig. 5. TPR and FPR for different k and N where threshold = 90%.

TABLE VII

DEFINITIONS OF VARIOUS EVALUATION METRICS

Accuracy	(TP+TN)/(TP+FP+TN+FN)
Recall	TP/(TP+FN)
Specificity	TN/(FP+TN)
Precision	TP/(TP+FP)
F-measure	2*Precision*Recall/(Precision+Recall)
G-mean	$(\text{Recall*Specificity})^{0.5}$

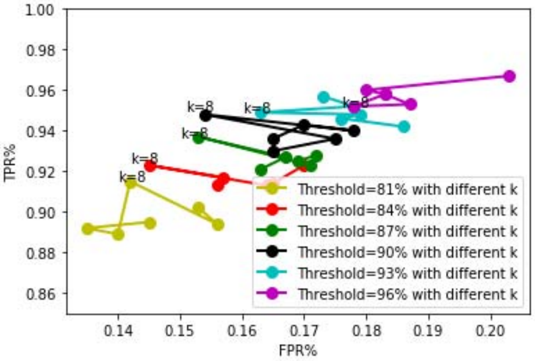


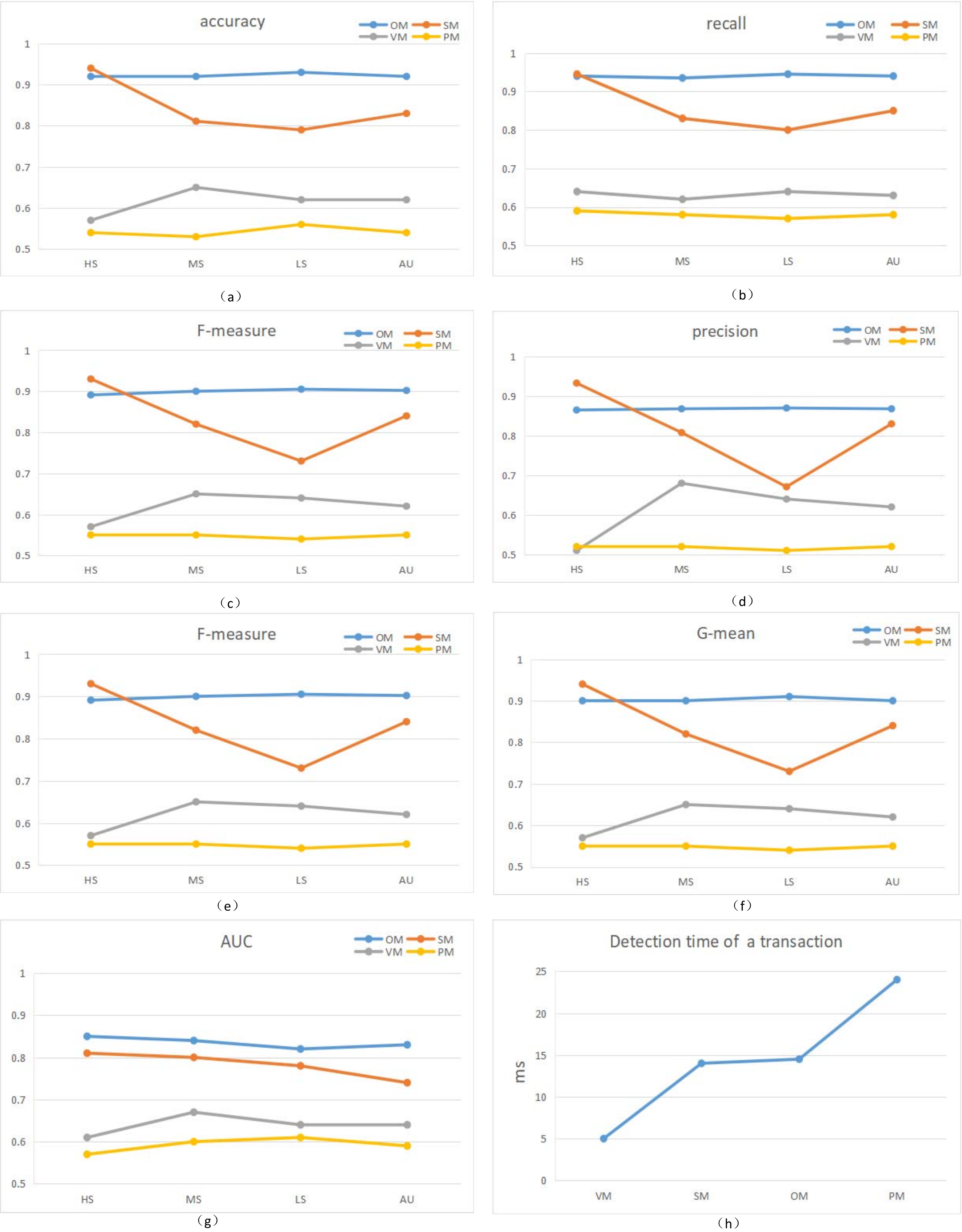
Fig. 6. TPR and FPR for different threshold and k where $N = 40$.

The transaction records are classified into three categories based on transaction amount: low amount (LA), medium amount (MA), and high amount (HA). If the transaction amount in a transaction record is in (0, 200], then the transaction record belongs to LA. If a transaction amount is in (200, 500], then the related record belongs to MA. If a transaction amount is greater than 500, then the transaction record is put into HA.

Consequently, we divide users into three groups: high stability (HS), medium stability (MS), and low stability (LS), in view of the distribution of a user's transaction records in LA, MA, and HA. If over 90% of a user's transaction records belong to LA, or over 90% belong to MA, or over 90% belong to HA, then the user is in HS. If the numbers of a user's records belonging to, respectively, LA, MA, and HA are all larger than or equal to 30% of her/his total records (and thus are all less than 40%), then the user is put into LS. For other cases, a related user is in MS. Obviously, for the users in MS and LS, their consume money are often changed, but for the users in HS, their consume money are not changed too much. Next, we respectively do experiments over HS users, MS users, LS users, and all users (AU). Note that the reason why we do experiments over these different groups is that: SM method in [33] is based on Markov chain, and thus is more suitable for users whose transaction amounts are not changed too much, i.e., those users in HS. In fact, our experiments also show that the SM method is better than ours for HS, but for other cases, OM is better than SM.

In order to provide a broad perspective on comparison, we use six popular evaluation metrics, which are listed in Table VII. TP is the total number of fraud transactions caught by the system. FP (False positive) is the total number of legal transactions labeled as fraudulent. True negative (TN) is the total number of legal transactions labeled as legal. False negative (FN) is the total number of fraud transactions which are not detected. Accuracy represents the percentage of the total number of transactions (both legal and fraudulent) which have been detected correctly. Recall and specificity measure the accuracy on the fraudulent and legal transactions, respectively. Precision gives the accuracy on transactions predicted as fraudulent. F-measure gives the harmonic mean of precision and recall, and G-mean gives the geometric mean of fraud and non-fraud accuracies. In addition, we consider Area Under Curve (AUC) performance measure, which is often considered as a better measure of overall performance [21]. AUC measures the area under the receiver operating characteristic curve and is independent on specific classification cutoff values.

The performance of these methods is shown in Fig. 7(a)–(g). The results show that PM and VM are inferior to OM and SM. From Fig. 7(a)–(f), we can see that for HS users, SM method is better than ours since SM is based on Markov chain, and thus is more suitable for the stable case. But for MS, LS, and AU users, OM is better than SM since our model considers the diversity of users' behaviors. From the AUC in Fig. 7(g), we also know that the overall performance of OM is better than the SM method. These experiments are done by a computer with a 2.8-GHz Intel i7 processor. As shown in Fig. 7(h), the detection time of OM is approximately equal to the detection time of



（a）

（b）

（c）

（d）

（e）

（f）

（g）

（h）

Fig. 7. Performance comparison of the four methods.

SM, more than the detection time of VM, but less than the detection time of PM.

V. Conclusion

In this paper, we propose a method to extract users' BPs based on their transaction records, which is used to detect transaction fraud in the online shopping scenario. OM overcomes the shortcoming of Markov chain models since it characterizes the diversity of user behaviors. Experiments also illustrate the advantage of OM. The future work focuses on some machine-learning methods to automatically classify the values of transaction attributes so that our model can characterize the user's personalized behavior more precisely. In addition, we plan to extend BP by considering other data such as user's comments.

Acknowledgment

The authors would like to thank the reviewers for their constructive comments.

References

[1] W. van der Aalst, T. Weijters, and L. Maruster, "Workflow mining: Discovering process models from event logs," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 9, pp. 1128–1142, Sep. 2004.

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

[3] N. Abdelhamid, A. Ayesh, and F. Thabtah, "Phishing detection based associative classification data mining," *Expert Syst. Appl.*, vol. 41, no. 13, pp. 5948–5959, 2014.

[4] N. M. Adams, D. J. Hand, G. Montana, D. J. Weston, and C. W. Whitrow, "Fraud detection in consumer credit," *Autumn*, vol. 9, no. 1, pp. 21–29, 2006.

[5] C. Arun, "Fraud: 2016 & its business impact," Assoc. Certified Fraud Examiners, Austin, TX, USA, Tech. Rep., Nov. 2016.

[6] A. Azaria, A. Richardson, S. Kraus, and V. S. Subrahmanian, "Behavioral analysis of insider threat: A survey and bootstrapped prediction in imbalanced data," *IEEE Trans. Comput. Social Syst.*, vol. 1, no. 2, pp. 135–155, Jun. 2014.

[7] V. Bhusari and S. Patil, "Application of hidden Markov model in credit card fraud detection," *Int. J. Distrib. Parallel Syst.*, vol. 2, no. 6, pp. 203–210, 2011.

[8] R. Brause, T. Langsdorf, and M. Hepp, "Neural data mining for credit card fraud detection," in *Proc. IEEE Int. Conf. Tools Artif. Intell.*, 1999, pp. 103–106.

[9] T. Carter, *An Introduction to Information Theory and Entropy*, S. Fe, Eds. CiteSeer, 2007.

[10] R. C. Chen, S. T. Luo, X. Liang, and V. C. S. Lee, "Personalized approach based on SVM and ANN for detecting credit card fraud," in *Proc. Int. Conf. Neural Netw. Brain*, Oct. 2005, pp. 810–815.

[11] C. Cortes and D. Pregibon, "Signature-based methods for data streams," *Data Mining Knowl. Discovery*, vol. 5, no. 3, pp. 167–182, 2001.

[12] V. Dheepa and R. Dhanapal, "Behavior based credit card fraud detection using support vector machines," *ICTACT J. Soft Comput.*, vol. 4, no. 4, pp. 391–397, 2012.

[13] S. G. Fashoto, O. Owolabi, O. Adeleye, and J. Wandera, "Hybrid methods for credit card fraud detection using K-means clustering with hidden Markov model and multilayer perceptron algorithm," *Brit. J. Appl. Sci. Technol.*, vol. 13, no. 5, pp. 1–11, 2016.

[14] *Global Online Payment Methods: Full Year 2016*, GmbH & Co. KG, Berlin, Germany, Mar. 2016.

[15] S. Gordon and R. Ford, "On the definition and classification of cybercrime," *J. Comput. Virol.*, vol. 2, no. 1, pp. 13–20, 2006.

[16] S. Gupta and R. Johari, "A new framework for credit card transactions involving mutual authentication between cardholder and merchant," in *Proc. Int. Conf. Commun. Syst. Netw. Technol.*, Jun. 2011, pp. 22–26.

[17] P. Hoffman and B. Schneier, *Attacks on Cryptographic Hashes in Internet Protocols*, document RFC 4270, 2005.

[18] 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 Things J.*, to be published, doi: 10.1109/JIOT.2018.2816007.

[19] W.-H. Ju and Y. Vardi, "A hybrid high-order Markov chain model for computer intrusion detection," *J. Comput. Graph. Stat.*, vol. 10, no. 2, pp. 277–295, 2004.

[20] A. Kundu, S. Panigrahi, S. Sural, and A. K. Majumdar, "BLAST-SSAHA hybridization for credit card fraud detection," *IEEE Trans. Dependable Secure Comput.*, vol. 6, no. 4, pp. 309–315, Oct. 2009.

[21] C. X. Ling, J. Huang, and H. Zhang, "AUC: A statistically consistent and more discriminating measure than accuracy," in *Proc. Int. Joint Conf. Artif. Intell.*, vol. 3, 2003, pp. 519–524.

[22] J. Lopes, O. Belo, and C. Vieira, "Applying user signatures on fraud detection in telecommunications networks," in *Proc. Ind. Conf. Data Mining*, 2011, pp. 286–299.

[23] S. S. Mhamane and L. M. R. J. Lobo, "Internet banking fraud detection using HMM," in *Proc. 3rd Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT)*, Jul. 2012, pp. 1–4.

[24] G. Mota, J. Fernandes, and O. Belo, "Usage signatures analysis an alternative method for preventing fraud in e-commerce applications," in *Proc. Int. Conf. Data Sci. Adv. Anal.*, Oct. 2014, pp. 203–208.

[25] S. Panigrahi, A. Kundu, S. Sural, and A. K. Majumdar, "Credit card fraud detection: A fusion approach using Dempster–Shafer theory and Bayesian learning," *Inf. Fusion*, vol. 10, no. 4, pp. 354–363, 2009.

[26] T. Patel and M. O. Kale, "A secured approach to credit card fraud detection using hidden Markov model," *Int. J. Adv. Res. Comput. Eng. Technol.*, vol. 3, no. 5, pp. 1576–1583, 2014.

[27] C. Phua, V. Lee, K. Smith, and R. Gayler. (2010). "A comprehensive survey of data mining-based fraud detection research." [Online]. Available: http://arxiv.org/abs/1009.6119

[28] J. T. Quah and M. Sriganesh, "Real-time credit card fraud detection using computational intelligence," *Expert Syst. Appl.*, vol. 35, no. 4, pp. 1721–1732, 2008.

[29] M. Schonlau, W. DuMouchel, W. H. Ju, A. F. Karr, M. Theus, and Y. Vardi, "Computer intrusion: Detecting masquerades," *Stat. Sci.*, vol. 16, no. 1, pp. 58–74, 2001.

[30] L. Seyedhossein and M. R. Hashemi, "Mining information from credit card time series for timelier fraud detection," in *Proc. Int. Symp. Telecommun.*, Dec. 2010, pp. 619–624.

[31] A. Shen, R. Tong, and Y. Deng, "Application of classification models on credit card fraud detection," in *Proc. Int. Conf. Service Syst. Service Manage.*, Jun. 2007, pp. 1–4.

[32] E. Shi, Y. Niu, M. Jakobsson, and R. Chow, "Implicit authentication through learning user behavior," in *Proc. Int. Conf. Inf. Secur.*, 2011, pp. 99–113.

[33] A. Srivastava, A. Kundu, S. Sural, and A. K. Majumdar, "Credit card fraud detection using hidden Markov model," *IEEE Trans. Dependable Secure Comput.*, vol. 5, no. 1, pp. 37–48, Jan./Mar. 2008.

[34] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, "Cost-based modeling for fraud and intrusion detection: Results from the JAM project," in *Proc. Inf. Survivab. Conf. Expo.*, vol. 2, 2000, pp. 130–144.

[35] D. L. Talekar and K. P. Adhiya, "Credit card fraud detection using hmm and image click point authentication," *Int. J. Adv. Stud. Comput., Sci. Eng.*, vol. 4, no. 3, pp. 1–4, 2015.

[36] V. V. Vlasselaer *et al.*, "*APATE*: A novel approach for automated credit card transaction fraud detection using network-based extensions," *Decis. Support Syst.*, vol. 75, pp. 38–48, Jul. 2015.

[37] C. Whitrow, D. J. Hand, P. Juszczak, D. J. Weston, and N. M. Adams, "Transaction aggregation as a strategy for credit card fraud detection," *Data Mining Knowl. Discovery*, vol. 18, no. 1, pp. 30–55, 2009.

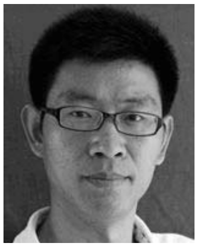
[38] V. Zaslavsky and A. Strizhak, "Credit card fraud detection using self-organizing maps," *Inf. Secur.*, vol. 18, no. 1, pp. 48–63, 2006.

[39] Y. Zheng, "An authentication and security protocol for mobile computing," in *Mobile Communications*. Boston, MA, USA: Springer, 1996, pp. 249–257.

[40] Z. Zojaji, R. E. Atani, and A. H. Monadjemi. (2016). "A survey of credit card fraud detection techniques: Data and technique oriented perspective." [Online]. Available: http://arxiv.org/abs/1611.06439

Lutao Zheng received the M.S. degree from the School of Computer and Information, Anhui Normal University, Wuhu, Anhui, China, in 2015. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology, Tongji University, Shanghai, China.

His current research interests include credit card fraud detection, machine learning, deep learning, big data, and transfer learning and natural language processing.



Guanjun Liu (M'16) received the Ph.D. degree in computer software and theory from Tongji University, Shanghai, China, in 2011.

From 2011 to 2013, he was a Post-Doctoral Research Fellow with the Singapore University of Technology and Design, Singapore. From 2013 to 2014, he was a Post-Doctoral Research Fellow with the Humboldt University of Berlin, Berlin, Germany, supported by the Alexander von Humboldt Foundation. He is currently an Associate Professor with the Department of Computer Science and Technology, Tongji University. He has authored over 70 papers including 13 ones in IEEE/ACM transactions and one book entitled *Liveness of Petri Nets and Its Application* (Tongji University Press, 2017). His current research interests include Petri net theory, model checking, Web service, workflow, discrete event systems, and information security.

Chungang Yan received the Ph.D. degree from Tongji University, Shanghai, China, in 2006.

She is currently a Professor with the Department of Computer Science and Technology, Tongji University. She has authored or co-authored more than 100 papers in domestic and international academic journals and conference proceedings. Her current research interests include concurrent model and algorithm, Petri net theory, formal verification of software, trusty theory on software process.

Changjun Jiang received the Ph.D. degree from the Institute of Automation, Chinese Academy of Science, Beijing, China, in 1995.

He is currently the Leader of the Key Laboratory of the Ministry of Education for Embedded System and Service Computing with Tongji University, Shanghai, China. He is a Honorary Professor with Brunel University London, London, U.K. He has authored or co-authored more than 300 papers in journals and conference proceedings. He has led over 30 projects. His current research interests include concurrence theory, Petri nets, formal verification of software, cluster, grid technology, intelligent transportation systems, and service-oriented computing.

Dr. Jiang is an IET Fellow. He was a recipient of one international prize and seven prizes in the field of science and technology.