

Received May 13, 2020, accepted May 25, 2020, date of publication May 28, 2020, date of current version June 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2998132

Vulnerability Evaluation Method for E-Commerce Transaction Systems With Unobservable Transitions

MIMI WANG¹, (Member, IEEE), ZHIJUN DING², (Senior Member, IEEE), AND PEIHAI ZHAO³, (Member, IEEE)

¹College of Information Science and Technology, Donghua University, Shanghai 201620, China

²Department of Computer Science and Engineering, Tongji University, Shanghai 201804, China

³School of Computer Science and Technology, Donghua University, Shanghai 201620, China

Corresponding authors: Zhijun Ding (dingzj@tongji.edu.cn) and Peihai Zhao (peihai.zhao@gmail.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB2100800, in part by the National Natural Science Foundation of China under Grant 61672381, in part by the Fundamental Research Funds for the Central Universities under Grant 22120180508 and Grant 2232020D-51, in part by the Shanghai Municipal Commission of Economy and Information Civil-Military Inoculation Project JMRH-2018-1042, and in part by the Initial Research Funds for Young Teachers of Donghua University.

ABSTRACT E-commerce transaction systems have become an important factor in trading activities. However, e-commerce systems are still undergoing development. Unobservable actions and attacks on systems are frequent problems that increase the vulnerability of e-commerce systems. Most existing approaches to addressing these issues cannot describe or analyze the overall structure of a local specification and unobservable actions well. The vulnerable e-commerce transaction net (VET-net) is a useful model for describing the unobservable actions, online transactions and third-party payment platforms of e-commerce systems. Based on a VET-net, we focus on the detection and evaluation of e-commerce transaction systems to attacks. We propose the concept of vulnerable transitions, which include not only vulnerable actions but also unobservable transitions. Then, we use an improved slice method to locate the vulnerable transitions. For these vulnerable transitions, we propose a vulnerable transition evaluation method based on a hidden Markov model along with a reachability graph (HMM-RG). The HMM-RG uses hidden Markov models (HMMs) to approximate the state reachability graph of a VET-net. By calculating the firing probability, the HMM-RG can evaluate the vulnerability degree of malicious states. We use a real-world case to show our method's effectiveness and reasonability.

INDEX TERMS E-commerce system, system vulnerability, vulnerability evaluation, labeled petri net, hidden Markov model.

I. INTRODUCTION

With the development of e-commerce, an increasing number of people are paying attention to the study of economic systems. Due to the imperfect nature of the e-commerce system itself, there are many problems in managing e-commerce systems [1]. There are some useful modeling methods to describe e-commerce systems, such as those of [2]–[8]. Du *et al.* [2] use a labeled Petri net (LPN) to analyze the obligation and accountability of cooperative systems and extend the LPN to a labeled workflow net (LWN) to model e-commerce workflows. However, they do not consider the

three parties (shopper, merchant, and third-party payment platform (TPP)) involved in the transaction. Yu *et al.* [4], [6] propose an e-commerce business process net (EBPN) to construct an e-commerce business process. Based on the EBPN, they describe malicious behavior patterns in [5]. The behavior patterns represent potential attacks that violate security [5]. However, they are not suitable for vulnerable e-commerce systems with unobservable actions.

For a safe system, there are methods that can be used to diagnose insecurity for users, such as those of [9]–[11]. In fact, this premise of system security is idealized. Most of the time, an e-commerce system is not completely secure. Thus, it is important to analyze vulnerable e-commerce systems. For a vulnerable e-commerce system, our previous

The associate editor coordinating the review of this manuscript and approving it for publication was Zhiwu Li¹.

work [12] proposes vulnerable e-commerce transaction nets (VET-nets). VET nets consider both normal actions and malicious actions. Malicious actions include attacks and unobservable actions. In this paper, we use VET-nets as modeling tools. For a VET-net, we know that if we provide a known attack form, we can locate the cause of the attack form in an e-commerce system using, e.g., the methods in [12]–[14]. Depending on the attack, there are vulnerable points used to complete the attack process. In fact, there may not be only one of these points. Some points are closely related to these unobservable transitions. These actions may play different roles in a VET-net. The different effects can be represented in terms of the effect degree. This effect degree is called the vulnerability degree. Existing methods mainly aim at observable system attacks, and they are suitable for vulnerability diagnosis from observable activities. For some unobservable activities and unpredictable attack forms, these methods are not applicable. From the perspective of observed actions, one only analyzes these unobservable states of malicious activity. Such a method ignores the effects of unobservable activities that lead to a malicious state. It cannot perform the appropriate adjustment of the e-commerce system. In other words, the system mostly performs analysis after malicious activity, and it does not prevent the effects. It is mostly postmortem analysis. It has difficulty preventing attacks. Another method only considers unexpected attack forms and ignores unobservable actions. It is not good for testing the e-commerce system itself. So our goal is to address two issues (i.e., unknown attacks and unobservable actions). The starting point of security practices for e-commerce systems is to evaluate their vulnerability. When a potential threat exploits the vulnerable actions of the system, it can lead to the destruction and damage of the system. Vulnerability assessment is a process of interpretation and vulnerability analysis. The purpose of vulnerability assessment is to find and control vulnerability. In fact, if we can control these unobservable actions and assess the occurrence probability of each path to the final condition, we can diagnose and assess the e-commerce system. Therefore, we emphasize unobservable actions.

Our goal is to determine how to use a formal method to identify the vulnerable actions of a VET-net system and to give a formal calculation method for the vulnerability degree of a VET-net. The contributions of this paper include:

- 1) We propose the concept of vulnerable transitions, which includes not only vulnerable actions but also unobservable transitions. We present a new method to locate the vulnerable transitions of a VET-net.
- 2) We use hidden Markov models (HMMs) to approximate the state reachability graph of a VET-net. Based on this, we give a vulnerable state evaluation method for a VET-net.

Fig. 1 shows the framework of our methods. In the first stage, by capturing the user transaction behaviors of the e-commerce trading process, we analyze the observable and unobservable actions and then construct a VET-net model.

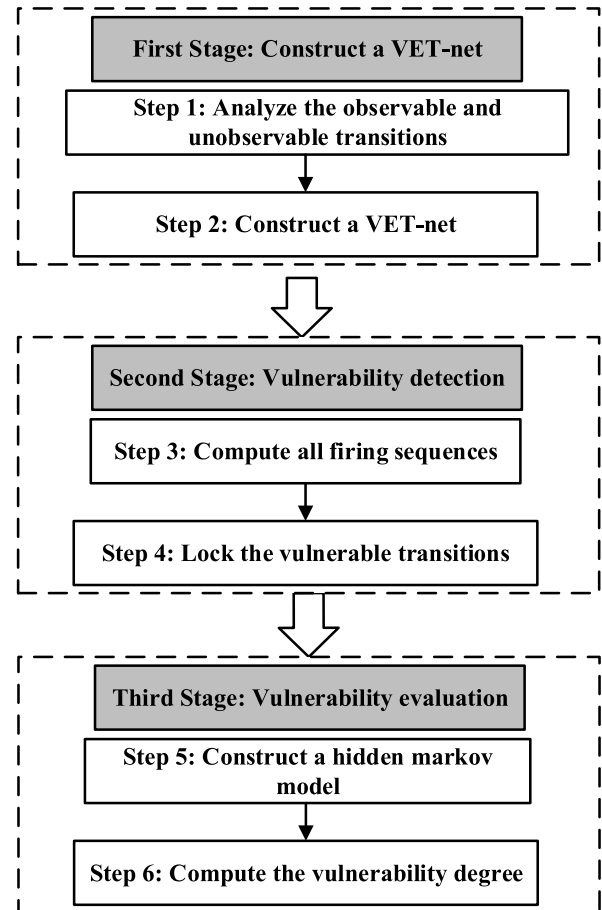


FIGURE 1. The framework of our methods.

Based on the VET-net, we can find all paths to the final condition; then, according to the characteristics of behaviors, we can locate the vulnerable actions of VET-nets, locating the source of fragility of the trading system for the user in the second step. In the third step, we can obtain a hidden Markov model according to the reachability graph of the VET-net. According to the hidden Markov model and reachability graph, we can compute the vulnerability degree of the VET-net.

The remainder of this paper is organized as follows: Section II discusses related work. Section III reviews some basic concepts and definitions as the basis of the study. Section IV proposes a vulnerability analysis and evaluation method. In Section V, we provide and analyze a real e-commerce example. Finally, we conclude in Section VI.

II. RELATED WORK

Some researchers have studied vulnerability detection and assessment. Some studies on vulnerability diagnosis, for example, Emeka and Liu [15] proposed a method to identify software security vulnerabilities. However, their work mostly concerned software rather than systems themselves. Xu and Nygard [7], Fang *et al.* [16] proposed a method to locate change based on behavioral profiles. They determined the

different parts by comparing the differences between two models. However, this method was based on two models, and it was not suitable for a single model. In fact, in most cases, we do not know the meta-model. In our previous work [12], [13], we proposed a method to diagnose the vulnerability of a model. At the same time, we provided a method to locate the vulnerable points. However, this method could not evaluate these vulnerable points. Allodi and Massacci [17] considered two-stage attacks and escalated the attacks. The escalated attack could be performed by exploiting local vulnerabilities in the target. Wang *et al.* [18] proposed a method for fault diagnosis of a timed Petri net (TPN). They used a fault diagnosis graph to diagnose an observable TPN. Li and Hadjicostis [19] proposed a method for estimating the minimum initial marking in labeled Petri nets. They used this method to determine the minimum number of resources. Prakash *et al.* [20] proposed a method to perform online fault detection and isolate multiple faults. They analyzed global and local faults by using the global fault sensitivity signature matrix (GFSSM) and fault sensitivity signature matrix (FSSM). Lefebvre [21] proposed a diagnosis decision method by analyzing observation sequences. However, this method was suitable for observable actions.

Some studies on vulnerability diagnosis, for example, Al-Dwairi and Kamala [22] used security, privacy, design, and content to evaluate the quality of B2C e-commerce websites. They gave an evaluation pattern for B2C e-commerce. However, for an arbitrary e-commerce system, they did not give a clear determination and evaluation method. Wang *et al.* [23] proposed a vulnerability evaluation method based on the attack graph. This method could address low-complexity attack paths. However, it could not address attack paths with unobservable activities. Fonseca *et al.* [24] gave a method and a tool to evaluate security mechanisms. This method could determine the possibility of injecting realistic vulnerabilities in a website. However, this diagnostic method was suitable for the process model itself. Pedroni [25] analyzed uncertainty modeling and quantification methods to assess reliability and risk. The analysis compared advanced methods for the modeling, simulation and analysis of safety-critical systems and infrastructure under uncertainty. Khalid *et al.* [26] examined the degree of customer satisfaction with an e-commerce system. They used a modified American customer satisfaction index (ACSI) model to describe 149 online data points. They showed that customer expectations and e-commerce service quality could affect perceived value. Najafi [27] introduced e-trust building models and provided a method to enhance e-commerce security. Grejner-Brzezinska *et al.* [28] proposed a method based on spatial positioning, navigation, and timing (PNT) to analyze accuracy, continuity, and reliability. Cabasino *et al.* [29] used Petri nets based on the notions of minimal explanations and markings to diagnose DESs. They assumed that fault events could be modeled by observable transitions. This method was suitable for observable transitions. Hu *et al.* [30] proposed a method based on active diagnosis to enhance the

diagnosability for a finite state automaton. This method had the computational advantage than the diagnoser-based ones, but it was not suitable for unobservable transitions. Basile *et al.* [31] proposed a state estimation and fault diagnosis method for a labeled-time Petri net. They used a modified state class graph (MSCG) to perform fault diagnosis. Shoukry *et al.* [32] proposed a method to evaluate security. They used satisfiability modulo theory to address the complexity of secure state estimation. Bonhomme [33] proposed a method to assess the marking of an unlabeled P-time Petri net with unobservable transitions. This method used the candidate firing sequences to estimate the marking. Koga *et al.* [34] used the full-state and associated output feedback control law to control and estimate the one-phase Stefan problem. Hu *et al.* [35] proposed a method to determine the optimal marked signal distribution. They used a given distortion constraint and expected embedding rate to obtain the optimal distribution.

Although these methods have their own advantages, they are not suitable for detecting the vulnerabilities of unobservable actions. Overall, no sufficient vulnerability assessment methods exist for VET-nets. Thus, in this paper, we focus on vulnerability assessment methods for VET-nets.

III. PRELIMINARIES

This section describes the basic concepts and definitions used in this paper. For more details, the definitions of Petri nets and labeled Petri nets can be found in [16], [36]–[47]. For the definitions of the hidden Markov model and Bayes' theorem, we can refer to [48]–[58].

The triple $N = (P, T, F)$ is a *net*, if it satisfies the conditions:

- 1) $P \cup T \neq \emptyset$;
- 2) $P \cap T = \emptyset$;
- 3) $F \subseteq ((P \times T) \cup (T \times P))$; and
- 4) $dom(F) \cup cod(F) = P \cup T$

where $dom(F) = \{x \in P \cup T \mid \exists y \in P \cup T : (x, y) \in F\}$ and $cod(F) = \{x \in P \cup T \mid \exists y \in P \cup T : (y, x) \in F\}$

P and T are two disjoint sets called the set of places and set of transitions, respectively. F is the flow relation of N .

A *Petri net* satisfies the following enabling and firing rules:

i) A transition $t \in T$ is *enabled* at M , denoted by $M[t]$, if $\forall p \in \cdot t : M(p) \geq 1$;

ii) *Firing* an enabled transition t yields a new marking M' , denoted by $M[t]M'$, where

$$M'(p) = \begin{cases} M(p) + 1, & \text{if } p \in t' \setminus \cdot t \\ M(p) - 1, & \text{if } p \in \cdot t \setminus t' \\ M(p), & \text{otherwise} \end{cases}$$

iii) If there exist transitions t_1, t_2, \dots, t_k and markings M_1, M_2, \dots, M_k such that $M[t_1]M_1[t_2] \dots M_{k-1}[t_k]M_k$, then M_k is *reachable* from M . The set of all markings reachable from M is denoted by $R(M)$, and $M \in R(M)$. The set of all markings reachable from M_0 is denoted by $R(M_0)$; it is

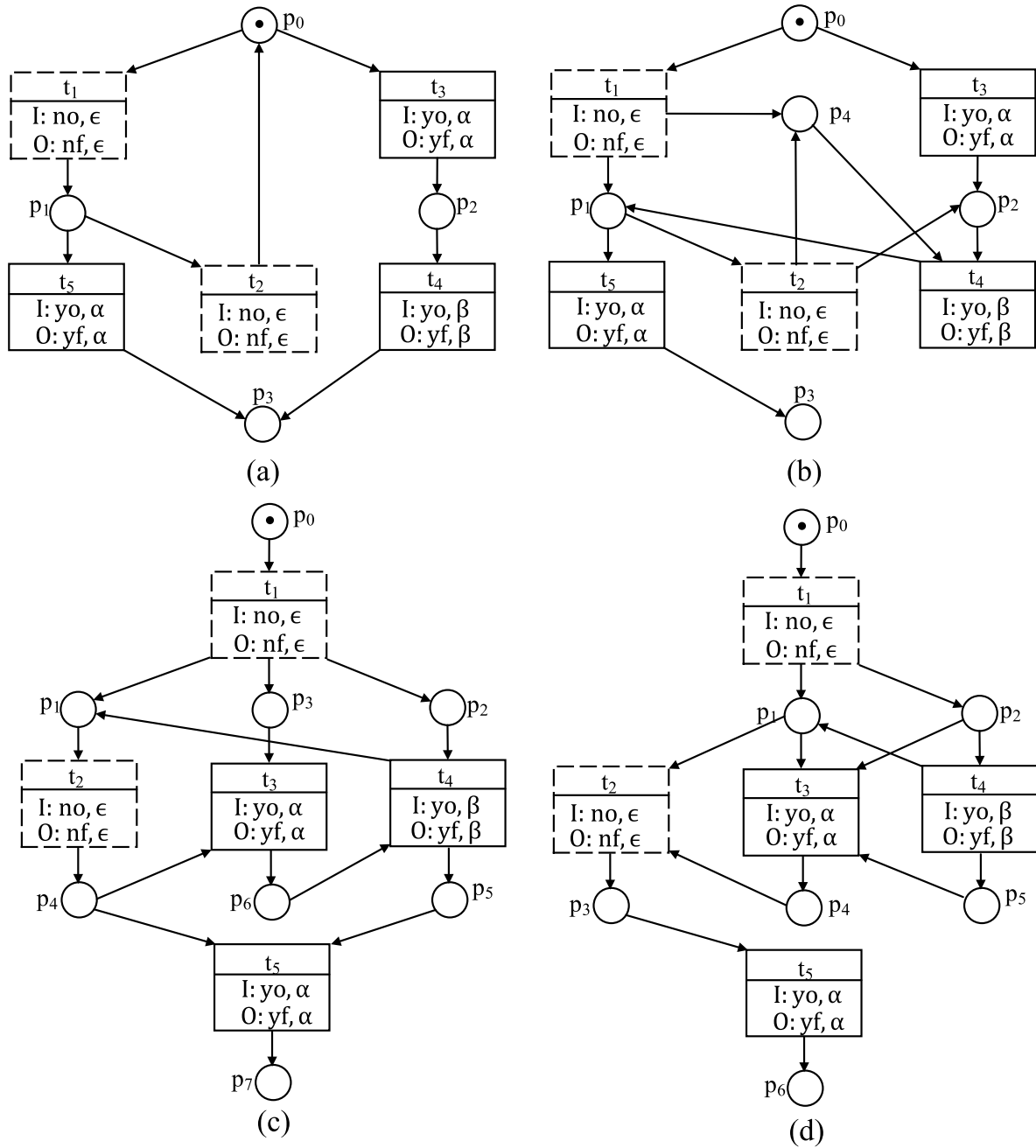


FIGURE 2. Four simple VET-nets.

called a *reachability marking set* and satisfies the following two conditions:

- 1) $M_0 \in R(M_0)$;
- 2) if $M \in R(M_0)$ and there is a $t \in T$ such that $M[t > M'$, then $M' \in R(M_0)$.

A *labeled PN* is a 3-tuple $G_L = (N, \Sigma, L)$, which is a labeling function that assigns a label (which can be the null label ϵ) to each transition.

To better describe the abnormal actions containing unobservable behaviors in the electronic transaction process, our

previous work [12] proposes a vulnerable e-commerce transaction net (VET-net).

Definition 1 (VET-net [12]): A VET-net is a 10-tuple $(P, T, F, M_0, p_I, p_F, A, I, O, \lambda)$, where:

- 1) (P, T, F, M_0) is a Petri net; P is a finite set of places, $p_I \in P$ is a source place satisfying $p_I = \emptyset$ and $p_F \in P$ is a sink place satisfying $p_F = \emptyset$;
- 2) A is a finite set of actions;
- 3) $I = \{yo, no, \epsilon\}$ is the set of input symbols;
- 4) $O = \{yf, nf, \epsilon\}$ is the set of output symbols;
- 5) $\lambda : T \rightarrow (A \times I \times O)$ is the label function.

where yo represents an observable input, no represents an unobservable input, yf represents an observable output, and nf represents an unobservable output.

The dotted box represents an unobservable transition. The dashed-line box represents an observable transition. Fig. 2 shows four simple VET-nets. In Fig. 2(a), the observable transitions are $t_3 : [I : yo, \alpha; O : yf, \alpha]$, $t_4 : [I : yo, \beta; O : yf, \beta]$, $t_5 : [I : yo, \alpha; O : yf, \alpha]$. The unobservable transitions are $t_1 : [I : no, \epsilon; O : nf, \epsilon]$, $t_2 : [I : no, \epsilon; O : nf, \epsilon]$. The enabling and firing rules can refer to GSPN [36] and literature [12].

In fact, from the definition of VET-net and GSPN, it is not difficult to see the difference between the two is that VET-net focuses on considering both normal actions and malicious actions. Hence, it is possible to use the PIPE tool to analyze its firing sequence and reachability analysis.

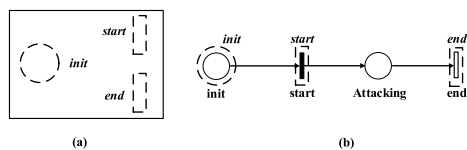


FIGURE 3. The attack presentation: (a) the attack' interface; (b) the attack' pattern [13].

An attack [59]–[62] is a sequence of actions $a_1.a_2 \dots a_n \in T$ such that there exists an elementary path from some initial state induced by $a_1 \dots a_n$ that reaches the set G . Fig. 3 characterizes the attack. Based on the attack, vulnerable points can be obtained. Then, the vulnerable transitions can be computed.

A vulnerable point [12] is a point that induces an attack. Accordingly, a vulnerable transition is a transition triggered by a vulnerable point and an unobservable transition.

For example, in Fig. 2(a), the unobservable transitions t_1, t_2, t_3 are the vulnerable transitions. If this VET-net is under an attack that induces state p_5 , all transitions triggered by p_5 are vulnerable transitions.

The hidden Markov model (HMM) [63] is a double random process: one process is a finite state Markov chain, which describes the state transitions; another stochastic process describes the statistical relations between the observed values. The state values and observed values obey a certain probability distribution. The hidden Markov model can be represented as a five-tuple $(\Omega_X, \Omega_O, A, B, \pi)$, where:

- 1) $\Omega_X = \{X_1, \dots, X_N\}$ is the finite set of hidden states, and N is the number of hidden states.
- 2) $\Omega_O = \{O_1, \dots, O_M\}$ is the finite set of observed values, and M is the number of observed values.
- 3) $A = \{a_{ij}\}$, $i, j = 1, 2, \dots, n$ is the state transition probability matrix, and $a_{ij} = P(X_{t+1} = q_j | X_t = q_i)$ denotes the transition probability from state q_i to q_j at moment t .
- 4) $B = \{b_{ij}(k)\}$ is the probability distribution of an observation, where b_{ij} denotes the probability of observation k in the transition from state q_i to q_j .

- 5) $\pi = \{\pi_1, \dots, \pi_n\}$, $\pi_1 = p(X_1 = q_j)$ is the initial state distribution.

For 1), we use s_t to denote the hidden state at time t , and the value of s_t is an element in the set Ω_X . For 4), let us suppose that $X = X_1, \dots, X_t$; then, $b_{ij}(k) = p(X_t = O_k | s_{t-1} = q_i, s_t = q_j)$.

IV. VULNERABILITY ANALYSIS AND EVALUATION

A. VULNERABILITY ANALYSIS

According to the definition of vulnerable transitions, if we locate the vulnerable points of the VET-net, we can compute the vulnerable transitions. We are inspired by the idea of slices [64], [65]. We assume that the malicious state is the final state, and the vulnerable points are the points in a slice of a VET-net that lead to malicious states. Algorithm 1 is the method to locate the vulnerable transitions.

According to steps 1-33 of Algorithm 1, we can obtain the set of vulnerable actions V_P . It contains three conditions, as follows:

- 1) For sequences $\sigma_1, \sigma_2, \dots, \sigma_n$, $\sigma_1 \cap \sigma_2 \cap \dots \cap \sigma_n \neq \emptyset$; Steps 1-13 describe this condition.
- 2) For sequences $\sigma_1, \sigma_2, \dots, \sigma_n$, there exists at most one firing sequence such that there is no other state except for P_q ; Steps 14-32 describe this condition.

Then, with steps 34-35, we can obtain all transitions of V_T .

Theorem 1: Algorithm 1 is correct and can be terminated.

Proof: Steps 1-33 absorb all conditions of firing path relations. If there is not only one path to P_q , then V_P is P_q , as shown steps 8, 18. If there is some common path C to P_q , then $V_P = slice(M_{i-1}, C, P_q)$ by step 6. Steps 34-35 satisfy the definition of vulnerable transitions. If there is another common path to P_q , then the set of vulnerable points is the whole set P , according the definition of slice, vulnerable point and vulnerable transition. Therefore, Algorithm 1 is correct. There are eight decision conditions, at Steps 4, 5, 7, 14, 15, 17, 24 and 26, that are used to determine the firing sequences. The firing sequences σ_i are finite sets. Then, the algorithm moves backward by adding a new firing sequence to the slice and removing the current slice until it is empty. Thus, Algorithm 1 will terminate. \square

Algorithm 1 is the process of locating vulnerable transitions. The input of Algorithm 1 consists of a VET-net, reachability graph, unobservable transitions set and goal state set. Given such input, it always terminates.

According to Algorithm 1, we can locate the vulnerable transitions of Fig. 2(a). First, we use PIPE tool¹ to obtain the VET-net and reachability graph of Fig. 2 as shown in Figs. 4 and 5. According to Fig. 5, we can see that there are paths $\sigma_1 = t_3t_4$, $\sigma_2 = t_1t_5$, \dots , $\sigma_i = (t_1t_2)^*t_3t_4$, $\sigma_j = (t_1t_2)^*t_1t_5$ to the state S_3 . $\sigma_1 \cap \sigma_2 \cap \dots \cap \sigma_i \cap \sigma_j = \emptyset$; there is not only one path to S_4 , and $\forall p \in P.M_{i-1}(p) \supseteq M_i(p)$. Then according to Algorithm 1, $V_P = slice(M_{i-1}, \bigcup_{i=1} \sigma_i, P)$. The vulnerable

¹<https://sourceforge.net/p/pipe2/bugs/milestone/PIPEv4.3.0/>

Algorithm 1 Vulnerable Transitions Location Algorithm

Input: VET-net, reachability graph RG , unobservable transitions set U_T , a goal state set P_q .
Output: Vulnerability transitions set V_T .

```

1 for all firing sequences  $\sigma_1, \sigma_2, \dots, \sigma_n$  from  $M_0$  to  $P_q$  do
2    $V_P = \emptyset$ ;
3    $V_T = \emptyset$ ;
4   if  $\sigma_1 \cap \sigma_2 \cap \dots \cap \sigma_n = C \neq \emptyset$  then
5     if  $\forall p \in P_q. M_{i-1}(p) \geq M_i(p)$  then
6        $V_P = \text{slice}(M_{i-1}, C, P_q)$ ;
7     else if  $i = 0$  then
8        $V_P = P_q$ ;
9     else
10       $V_P = \{t_i\} \cup \text{slice}(M_{i-1}, C, P_q \cup t_i)$ ;
11    end
12  end
13  if  $\sigma_1 \cap \sigma_2 \cap \dots \cap \sigma_n = \emptyset$  and  $\nexists \sigma_j (\sigma_j \neq \sigma_i)$  s.t.,
14   $\sigma_j \rightarrow P_q \wedge \sigma_i \rightarrow P_q$  then
15    if  $\forall p \in P_q. M_{i-1}(p) \geq M_i(p)$  then
16       $V_P = \text{slice}(M_{i-1}, \sigma_i, P_q)$ ;
17    else if  $i = 0$  then
18       $V_P = P_q$ ;
19    else
20       $V_P = \{t_i\} \cup \text{slice}(M_{i-1}, \sigma_i, P_q \cup t_i)$ ;
21    end
22  else
23    if  $\forall p \in P. M_{i-1}(p) \geq M_i(p)$  then
24       $V_P = \text{slice}(M_{i-1}, \bigcup_{i=1}^n \sigma_i, P)$ ;
25    else if  $i = 0$  then
26       $V_P = P$ ;
27    else
28       $V_P = \{t_i\} \cup \text{slice}(M_{i-1}, \bigcup_{i=1}^n \sigma_i, P \cup t_i)$ ;
29    end
30  end
31 end
32 end
33 end
34 for all transitions of  $V_P$ , compute all unobservable
35 transitions set  $U_T$  do
36    $V_T = \{t | p_i, p_i \in V_P\} \cup U_T$ 
37 end

```

transitions set is as follows:

$$\begin{aligned}
 V_T &= \{t | p_i, p_i \in V_P\} \cup U_T \\
 &= \{t_1, t_2, t_3, t_4, t_5\} \cup \{t_1, t_2\} \\
 &= \{t_1, t_2, t_3, t_4, t_5\}.
 \end{aligned} \tag{1}$$

Property 1: It takes polynomial time to compute V_T of a bounded VET-net.

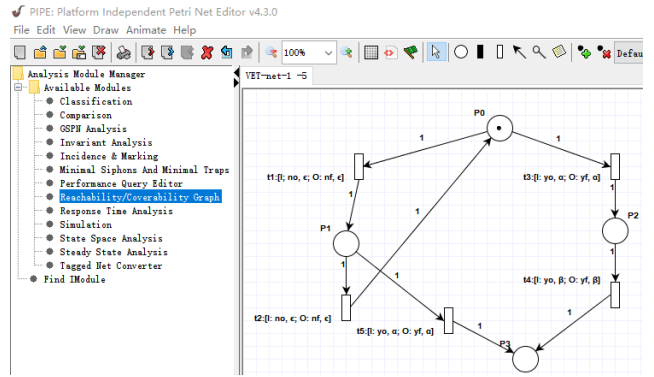


FIGURE 4. Fig. 2(a) in PIPE tool.

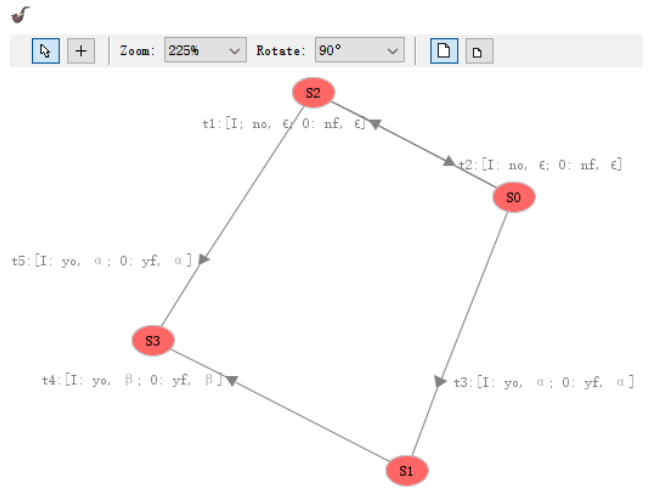


FIGURE 5. The reachability graph of Fig. 2(a).

Proof: For a bounded VET-net, the number of places, transactions, arcs and firing sequences are all finite. Let $|P|$ be the number of places, $|T|$ be the number of transitions, $|F|$ be the number of arcs in the VET-net and $|\sigma|$ be the number of firing sequences. Then, we can see that most of the time is spent computing vulnerable transitions. In each sequence, there are at most $|T|$ transactions and at most $|T| + 1$ states. Therefore, the first step computes the first sequence σ_1 , which requires $|\sigma| - 1$ operations. The second step computes the sequence σ_2 , which costs $|\sigma| - 2$ operations, and the third step computes the sequence σ_3 , which requires $|\sigma| - 3$ operations. We continue with similar calculations until the last sequence $\sigma_{|\sigma|}$. Thus, the total computing time is $[(|\sigma| - 1)|\sigma|/2]$. In addition, each sequence has at most $|T| + 1$. And we can see that the time of computing unobservable transitions at most need spent $|T|$. Hence, the V_T can be constructed in polynomial time (i.e., $O([(|T| + 1)(|\sigma| - 1)|\sigma|/2])$). \square

Similarly, for Fig. 2(c), the vulnerable transitions set is as follows:

$$\begin{aligned}
 V_T &= \{t | p_i, p_i \in V_P\} \cup U_T \\
 &= \{t_1, t_2, t_3, t_4, t_5\} \cup \{t_1, t_2\} \\
 &= \{t_1, t_2, t_3, t_4, t_5\}.
 \end{aligned} \tag{2}$$

In fact, we find that the vulnerable transitions sets of Figs. 2(a), (b), (c) and (d) are the same set $\{t_1, t_2, t_3, t_4, t_5\}$. In Fig. 2(c), the vulnerable transition t_2 occurs two times. However, in Figs. 2(b) and (d), the vulnerable transition t_2 occurs only one time. To distinguish these conditions, we give the evaluation method for these vulnerable transitions in section IV. B.

B. VULNERABILITY EVALUATION

The whole computation process is as follows: 1) use the PIPE tool to obtain the reachability graph; 2) compute the occurrence probability of vulnerable transitions based on the hidden Markov model (HMM) along with the reachability graph (HMM-RG). 3) use the HMM-RG state occurrence probability to obtain the evaluation computation of vulnerable transitions.

Given a VET-net $(P, T, F, M_0, p_I, p_F, A, I, O, \lambda)$ with initial markings in the finite set $\mathcal{M}_0 = M_0^{(1)}, M_0^{(2)}, \dots, M_0^{(M_0)}$ and an observed label sequence $\omega \in \Sigma^*$ (i.e., $\omega \in \Sigma^*$, $\omega \neq \varepsilon$), according to [66], we know a *a priori probability* for each initial marking, i.e., $Pr(M_0^{(i)}) = p_0^{(i)}$, $i = 1, 2, \dots, |\mathcal{M}_0|$, where $\sum_i p_0^{(i)} = 1$. Given a sequence of observations ω , along with their a *posteriori probabilities*, we can obtain

$$\mathcal{C}_r(\omega) = \{M \in \mathbb{N}^n \mid \exists S \in T^*T_o, \exists M_0 \in \mathcal{M}_0 : \\ \{M_0[S]M, L(S) = \omega\}\} \quad (3)$$

$$= \{(M^{(l)}, p^{(l)}(\omega)) \mid M^{(l)} \in \mathcal{C}_r(\omega)\}, \quad (4)$$

where

$$\mathcal{S}_r(\omega) = \{S \in T^*T_o \mid \exists M_0 \in \mathcal{M}_0 : \{M_0[S], L(S) = \omega\}\}. \quad (5)$$

$p^{(l)}(\omega) = Pr(M^{(l)} \mid \omega)$ is a posteriori probability. $\sum_{l: M^{(l)} \in \mathcal{C}_r(\omega)} p^{(l)}(\omega) = 1$. For $\omega = \varepsilon$, $\mathcal{S}_r(\varepsilon) = \varepsilon$ and $\mathcal{C}_r(\varepsilon) = \mathcal{M}_0$.

We will denote these probabilities by $Pr(k \mid j) \equiv Pr(s^{(k)} \mid M_0^{(j)})$, $\forall M_0^{(j)} \in \mathcal{M}_0, s^{(k)} \in T^*$. Similarly, we use $Pr(k, j) \equiv Pr(s^{(k)}, M_0^{(j)}) = Pr(k \mid j)p_0^{(j)}$ to indicate that the VET-net started at the initial marking $M_0^{(j)}$ and the sequence of transitions $s^{(k)}$ occurred.

Given a sequence of observations ω , we can obtain the *conditional probabilities* $p^{(l)}(\omega)$, where l is an arbitrary index in $\mathcal{C}_r(\omega)$. The only possible firing sequences are sequences in the set $\mathcal{S}_r(\omega)$. Since the sequences in the set $\mathcal{S}_r(\omega)$ are not prefixes of each other, the probability of observing ω is

$$Pr(\omega) = \sum_{j, M_0^{(j)} \in \mathcal{M}_0} \sum_{k: s^{(k)} \in \mathcal{S}_r(\omega)} Pr(k, j) \\ = \sum_{j, M_0^{(j)} \in \mathcal{M}_0} p_0^{(j)} \sum_{k: s^{(k)} \in \mathcal{S}_r(\omega)} Pr(k \mid j) \quad (6)$$

The probability of observing ω is the sum of the joint probabilities $Pr(s^{(k)}, M_0^{(j)})$ [66]. For each marking

$M^{(l)} \in \mathcal{C}_r(\omega)$, we can calculate using Bayes' rule and (4):

$$p^{(l)}(\omega) \equiv Pr(M^{(l)} \mid \omega) = \frac{Pr(M^{(l)}, \omega)}{Pr(\omega)} \\ = \frac{\sum_{j, M_0^{(j)} \in \mathcal{M}_0} \sum_{k: s^{(k)} \in \mathcal{S}_r(\omega) \text{ and } M_0^{(j)}[S_K]M^{(l)}} Pr(k, j)}{\sum_{j, M_0^{(j)} \in \mathcal{M}_0} \sum_{k: s^{(k)} \in \mathcal{S}_r(\omega)} Pr(k, j)} \quad (7)$$

$$= \frac{\sum_{j, M_0^{(j)} \in \mathcal{M}_0} p_0^{(j)} \sum_{s^{(k)} \in \mathcal{S}_r(\omega): M_0^{(j)}[S_K]M^{(l)}} Pr(k \mid j)}{\sum_{j, M_0^{(j)} \in \mathcal{M}_0} p_0^{(j)} \sum_{k: s^{(k)} \in \mathcal{S}_r(\omega)} Pr(k \mid j)} \quad (8)$$

According to [66], we can obtain an associated probability $p_M(t)$ that indicates the a priori probability that t fires at M . In addition, $\sum_{t \in T: M(t)} p_M(t) < 1$. Given $s^{(k)} = t_{s_1} t_{s_2} \dots t_{s_l}$ such that $M_0^{(j)}[t_{s_1}]M_1[t_{s_2}]M_2 \dots [t_{s_l}]M_l$,

$$Pr(k, j) = \underbrace{p_0^{(j)}}_{Pr(M_0^{(j)})} \times \frac{1}{1 + |T_{M_i^{(j)}}|} \prod_{i=1}^{l-1} \frac{1}{1 + |T_{M_i}|}. \quad (9)$$

Then for $k \geq 2$, we have

$$\mathcal{C}_r(e_{i_1} e_{i_2} \dots e_{i_k}) = \{M \in \mathbb{N}^n \mid \exists S \in T^*T_o, \exists M_0 \in \mathcal{M}_r \text{ s.t.} \\ L(S) = e_{i_1} e_{i_2} \dots e_{i_k}, M_0[S]M\}, \quad (10)$$

$$\mathcal{C}_r(e_{i_1} e_{i_2} \dots e_{i_k}) = \{M \in \mathbb{N}^n \mid \exists S', S'' \in T^*T_o, \exists M_0 \in \mathcal{M}_r \\ \exists M' \in R(N, \mathcal{M}_r) \text{ s.t.} \\ L(S') = e_{i_1} e_{i_2} \dots e_{i_{k-1}}, \\ L(S'') = e_{i_k}, M_0[S']M', M'[S'']M\}. \quad (11)$$

We see that the marking M' in the above definition satisfies $M' \in \mathcal{C}_r(e_{i_1} e_{i_2} \dots e_{i_k})$. Thus, we have

$$\mathcal{C}_r(e_{i_1} e_{i_2} \dots e_{i_k}) = \{M \in \mathbb{N}^n \mid \exists S \in T^*T_o, \\ \exists M' \in \mathcal{C}_r(e_{i_1} e_{i_2} \dots e_{i_k}) \\ \text{s.t. } L(S) = e_{i_k}, M'[S]M\} \quad (12)$$

To obtain $\mathcal{C}_{rP}(e_{i_1} e_{i_2} \dots e_{i_k})$ by recursion based on $\mathcal{C}_{rP}(e_{i_1} e_{i_2} \dots e_{i_{k-1}})$, let $\omega' = e_{i_1} e_{i_2} \dots e_{i_{k-1}}$ and $\omega = e_{i_1} e_{i_2} \dots e_{i_k} = \omega' e_{i_k}$, we can focus on calculating the numerator of the expression in (7), i.e.,

$$p_u^{(l)}(\omega) = \sum_{j, M_0^{(j)} \in \mathcal{M}_0} p_0^{(j)} \sum_{s^{(k)} \in \mathcal{S}_r(\omega): M_0^{(j)}[s^{(k)}]M^{(l)}} Pr(k \mid j) \\ \equiv Pr(M^{(l)}, \omega), \quad (13)$$

where p_u stands for an *unnormalized probability*. Using this decomposition for each string $s^{(k)} \in \mathcal{S}_r(\omega)$, we can write the second sum as follows:

$$\sum_{s^{(k)} \in \mathcal{S}_r(\omega): M_0^{(j)}[s^{(k)}]M^{(l)}} Pr(k \mid j) \\ = \sum_{s^{(k')}, M^{(l')}: s: M_0^{(j)}[s^{(k')}]M^{(l')}[s]M^{(l)}} Pr(s \mid l') Pr(k' \mid j) \\ = \sum_{s^{(k')}, M^{(l')}: M_0^{(j)}[s^{(k')}]M^{(l')}} Pr(k' \mid j) \sum_{s: M^{(l')}[s]M^{(l)}} Pr(s \mid l'). \quad (14)$$

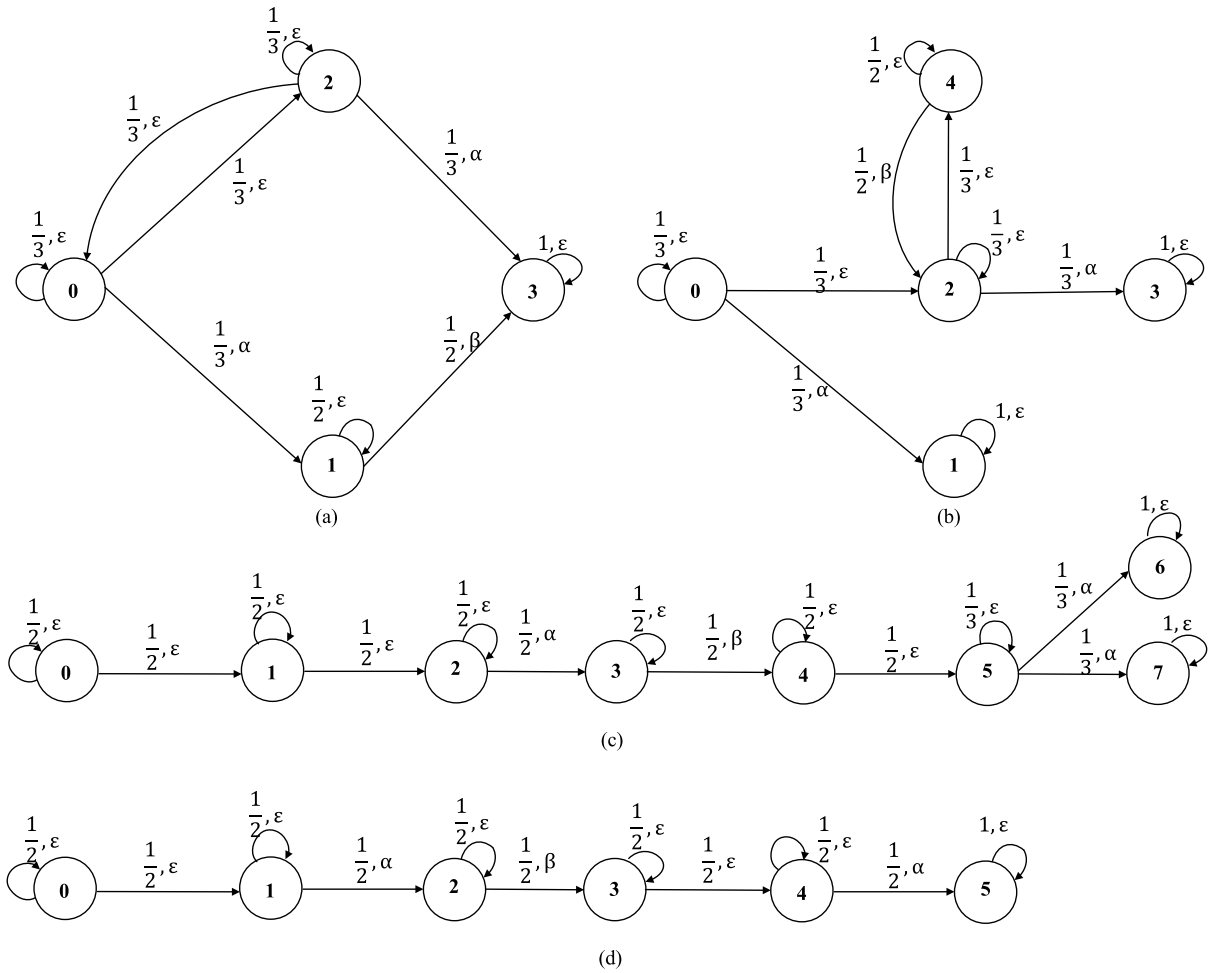


FIGURE 6. HMM of Fig. 5.

Then, according to (13) and (14), we have

$$p_u^{(l)}(\omega) = \sum_{M^{(l')} \in \mathcal{C}_r(\omega'), s \in \mathcal{S}_r(e_{ik}): M^{(l')}[s]M^{(l)}} p_u^{(l')}(\omega') P_r(s | l'). \tag{15}$$

We take Fig. 2 as an example. The Markov chain model of Fig. 5 is shown in Fig. 6. For Fig. 6(a), the reachable markings are: $M^{(0)} = [1\ 0\ 0\ 0]^T = M_0$, $M^{(2)} = [0\ 0\ 1\ 0]^T$, $M^{(2)} = [0\ 1\ 0\ 0]^T$, $M^{(3)} = [0\ 0\ 0\ 1]^T$.

For marking $M^{(0)}$, we have $Pr_{M^{(0)}}(t1) = Pr_{M^{(0)}}(t3) = Pr_{M^{(0)}}(\text{no firing}) = 1/3$. For marking $M^{(1)}$, we have $Pr_{M^{(1)}}(t4) = Pr_{M^{(1)}}(\text{no firing}) = 1/2$. For marking $M^{(2)}$, we have $Pr_{M^{(2)}}(t2) = Pr_{M^{(2)}}(t5) = Pr_{M^{(2)}}(\text{no firing}) = 1/3$. For marking $M^{(3)}$, there is no firing, and we have $Pr_{M^{(3)}} = 1$.

In Fig. 6(a), we know that observation α occurs via a transition sequence of $(t1t2)^*t3$ or $(t1t2)^*t5$, which leads to the marking $M^{(1)}$ or $M^{(3)}$. We can calculate the unnormalized probabilities $p_u^{(1)}(\alpha)$ and $p_u^{(3)}(\alpha)$ as follows:

$$p_u^{(1)}(\alpha) = \sum_{M^{(l')} \in \mathcal{C}_r(\omega'), s \in \mathcal{S}_r(e_{ik}): M^{(l')}[s]M^{(l)}} p_u^{(l')}(\omega') P_r(s | l')$$

TABLE 1. Normalized probabilities of Fig. 6.

Figure	Observation sequence	Set $\mathcal{C}_{r,P}$
Fig. 6(a)	ϵ	$\{(M^{(0)}, 1)\}$
Fig. 6(a)	α	$\{(M^{(1)}, 1/2), (M^{(3)}, 1/2)\}$
Fig. 6(a)	$\alpha\beta$	$\{(M^{(3)}, 1)\}$
Fig. 6(b)	ϵ	$\{(M^{(0)}, 1)\}$
Fig. 6(b)	α	$\{(M^{(1)}, 5/7), (M^{(3)}, 2/7)\}$
Fig. 6(b)	β	$\{(M^{(2)}, 1)\}$
Fig. 6(b)	$\beta\alpha$	$\{(M^{(3)}, 1)\}$
Fig. 6(c)	ϵ	$\{(M^{(0)}, 4/5), (M^{(2)}, 1/5)\}$
Fig. 6(c)	α	$\{(M^{(3)}, 1)\}$
Fig. 6(c)	$\alpha\beta$	$\{(M^{(4)}, 1)\}$
Fig. 6(c)	$\alpha\beta\alpha$	$\{(M^{(6)}, 1/2), (M^{(7)}, 1/2)\}$
Fig. 6(d)	ϵ	$\{(M^{(0)}, 1)\}$
Fig. 6(d)	α	$\{(M^{(2)}, 1)\}$
Fig. 6(d)	$\alpha\beta$	$\{(M^{(3)}, 1)\}$
Fig. 6(d)	$\alpha\beta\alpha$	$\{(M^{(5)}, 1)\}$

$$\begin{aligned}
 &= Pr(M^{(0)})(Pr(t3) + Pr((t1t2)t3) + Pr((t1t2)^2t3) + \dots) \\
 &= 1(1/3 + (1/3)^2(1/3) + (1/3)^4(1/3) + \dots) \\
 &= 1/3(1 + (1/3)^2 + (1/3)^4 + (1/3)^6 + \dots)
 \end{aligned}$$

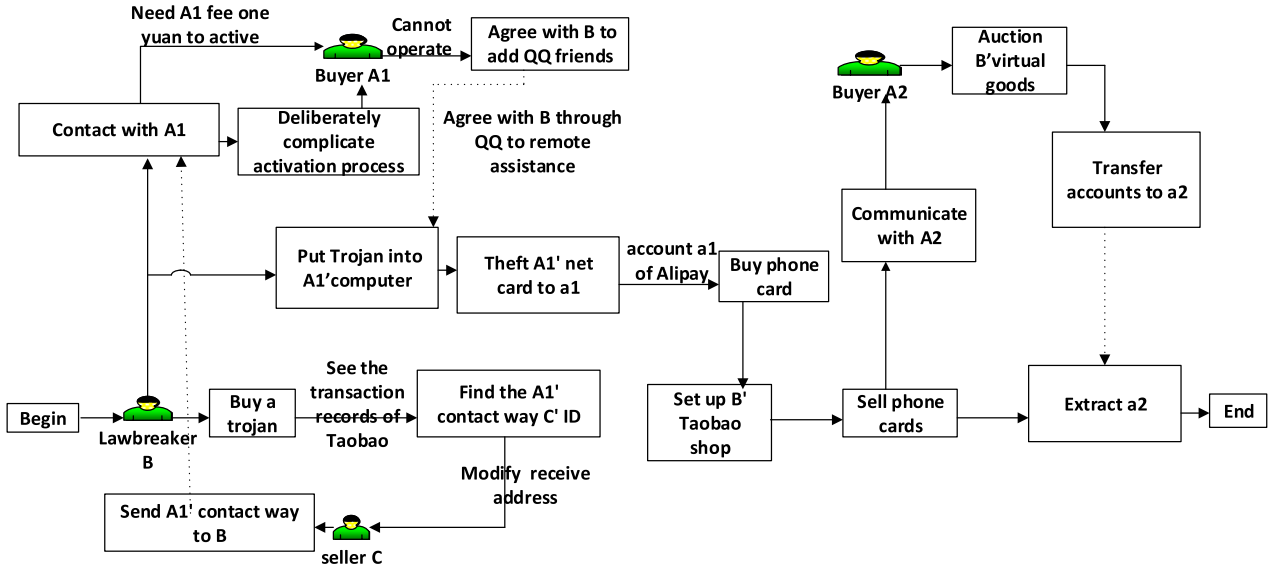


FIGURE 7. The pay case of business process [12].

TABLE 2. Transitions illustration [12] of Fig. 8 (✓ = Yes and × = No).

T	Meaning	Observable	Unobservable	Label
a_{11}	Receive B' call	✓	×	ϵ
a_{14}	Give the activation fee	✓	×	ϵ
a_{15}	Pay	✓	×	ϵ
a_{21}	Sell B' goods	✓	×	γ
a_{22}	Transfer to a2	✓	×	γ
b_{12}	Query the Taobao transaction records	✓	×	β
b_{11}	Buy a Trojan	✓	×	β
b_{13}	Find C' contact way and A1' ID	✓	×	β
b_{14}	Get in touch with A	×	✓	β
b_{15}	Need A1 fee one yuan to active	✓	×	β
b_{16}	Make the process complicated	✓	×	β
b_{17}	Apply QQ friends with A1	✓	×	β
b_{19}	Put Trojan	✓	×	β
b_{20}	Steal A1' net card a_1	×	✓	β
b_{22}	Build a Taobao shop	✓	×	β
b_{23}	Sell telephone cards	✓	×	β
b_{26}	Falsely claim to be A1, contact with C	×	✓	β
b_{27}	Obtain A1' information	✓	×	β
c_{11}	Receive the call of B	✓	×	α
c_{12}	Send A1' information to B	✓	×	α
c_{14}	Ask A1 pay	✓	×	α

$$\begin{aligned}
 &= 1/3 \times \lim_{n \rightarrow +\infty} \frac{(1 - (1/3)^{2n})}{1 - (1/3)^2} \\
 &= 3/8.
 \end{aligned} \tag{16}$$

$$\begin{aligned}
 &p_u^{(3)}(\alpha) \\
 &= \sum_{M^{(l)} \in \mathcal{C}_r(\omega'), s \in \mathcal{S}_r(e_{i_k}): M^{(l)}[s]M^{(l)}} p_u^{(l)}(\omega') P_r(s | l') \\
 &= Pr(M^{(0)})(Pr(t5) + Pr((t1t2)t5) + Pr((t1t2)^2t5) + \dots)
 \end{aligned}$$

$$\begin{aligned}
 &= 1(1/3 + (1/3)^2(1/3) + (1/3)^4(1/3) + \dots) \\
 &= 1/3(1 + (1/3)^2 + (1/3)^4 + (1/3)^6 + \dots) \\
 &= 1/3 \times \lim_{n \rightarrow +\infty} \frac{(1 - (1/3)^{2n})}{1 - (1/3)^2} \\
 &= 3/8.
 \end{aligned} \tag{17}$$

Similarly, we know that observation $\alpha\beta$ occurs via a transition sequence of $(t1t2)^*t3t4$, which leads to the marking $M^{(3)}$.

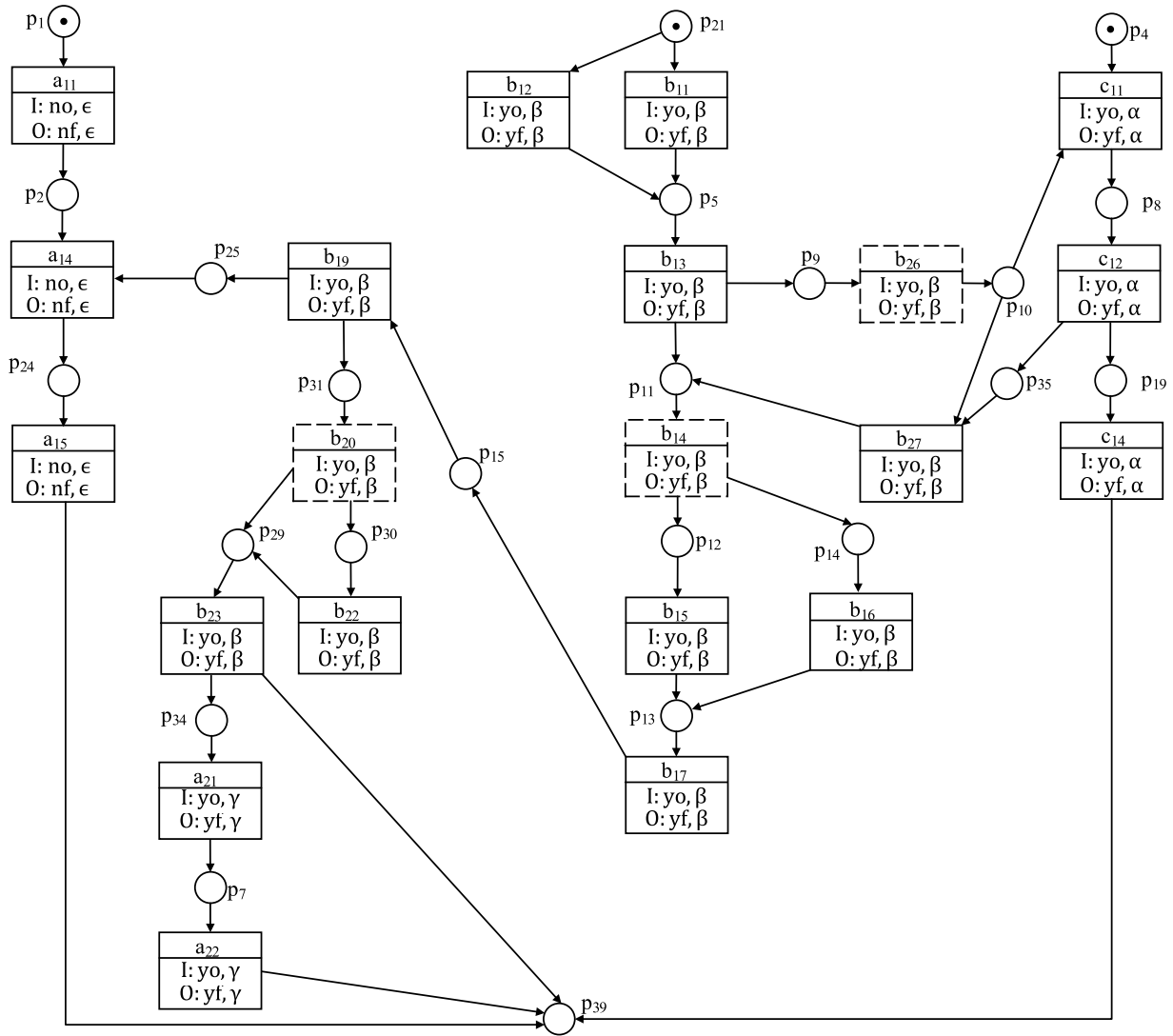


FIGURE 8. The VET-net model of Fig. 7.

We can calculate the unnormalized probability $p_u^{(3)}(\alpha\beta)$ as follows:

$$\begin{aligned}
 & p_u^{(3)}(\alpha\beta) \\
 &= \sum_{M^{(l)} \in \mathcal{C}_r(\omega'), s \in \mathcal{S}_r(e_{i_k}): M^{(l)}[s]M^{(l)}} p_u^{(l)}(\omega') P_r(s | l') \\
 &= Pr(M^{(0)})(Pr(t3t4) + Pr((t1t2)t3t4) + Pr((t1t2)^2t3t4) + \dots) \\
 &= 1(1/3 \times 1/2 + (1/3)^2(1/3 \times 1/2) + (1/3)^4(1/3 \times 1/2) + \dots) \\
 &= 1/6(1 + (1/3)^2 + (1/3)^4 + (1/3)^6 + \dots) \\
 &= 1/6 \times \lim_{n \rightarrow +\infty} \frac{(1 - (1/3)^{2n})}{1 - (1/3)^2} \\
 &= 3/16. \tag{18}
 \end{aligned}$$

The normalized probabilities of Fig. 6 are given in Table 1.

V. CASE STUDY

In 2015, a Tmall fraud case occurred in Mongolia in China. This process is shown in Fig. 7. We provide a description of this case [12]. First, *B* obtains the buyers' and sellers' contact information through a Trojan. Second, *B* tells seller *C* to help *A* change this information. Then, *B* implants a Trojan program into *A*'s computer and transfers *A*'s money to *a1*. Then, *B* uses *a1* to buy cards and builds a Tmall shop. *B* transfers money from *a1* to *a2* as in a normal transaction [12]. The VET-net model of Fig. 7 is shown in Fig. 8. Table 2 describes the meaning of the transitions in Fig. 8.

We use PIPE to construct a reachability graph, as shown in Fig. 9. Figs. 10 and 11 show the overview of the transition sequences. In Fig. 10, the horizontal axis shows each state, and the vertical coordinates show the number of sequences. According to 10, there are 504 firing sequences to reach state S_{47} . In Fig. 11, the horizontal axis shows the number of layers. Layer 1 indicates the first layer, including

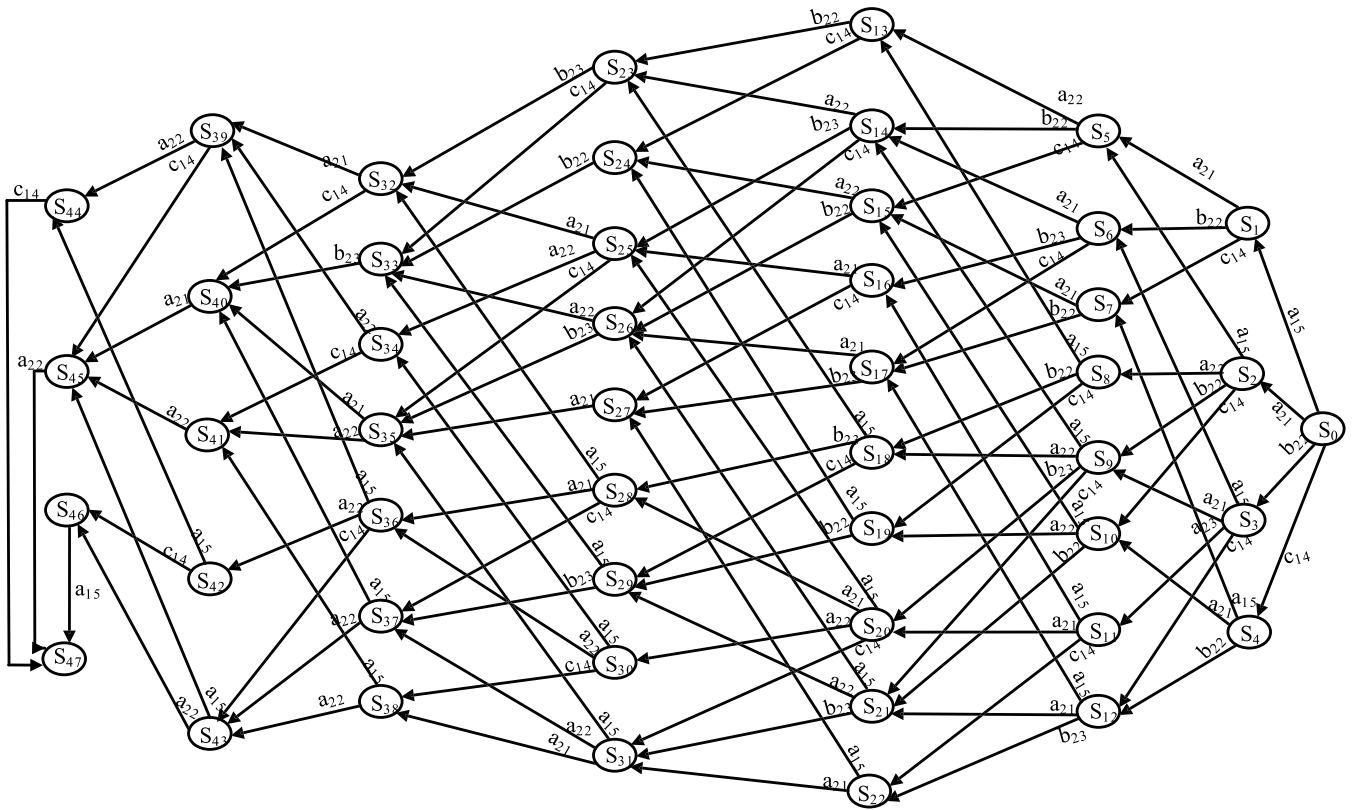


FIGURE 9. PIPE implementation result of reachability graph of Fig. 8.

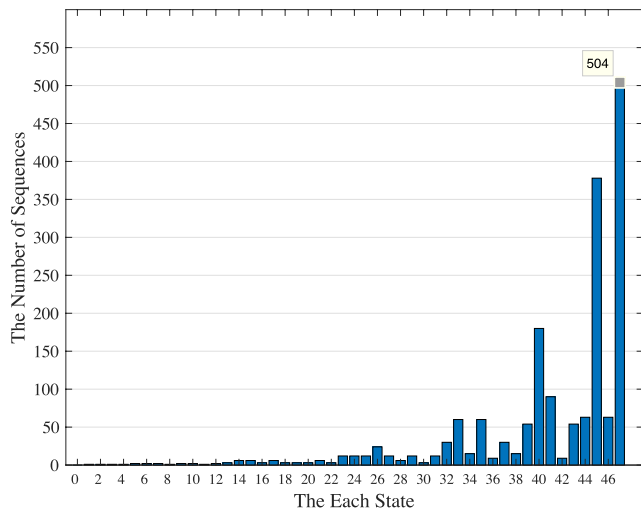


FIGURE 10. The number of all transitions sequences corresponding to each state.

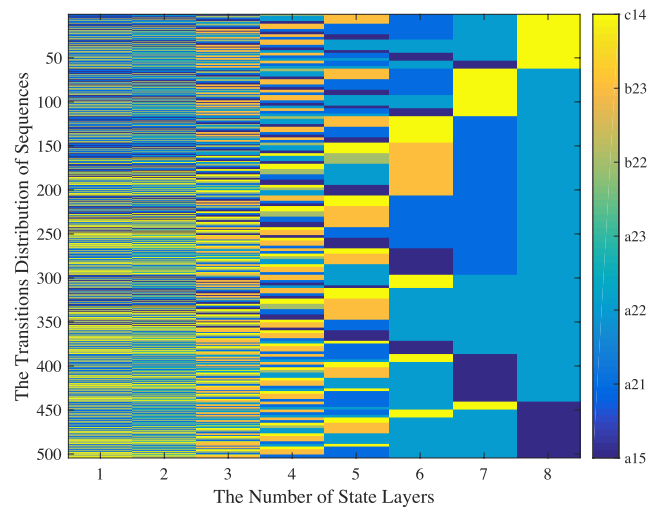


FIGURE 11. The overview distribution of the transitions sequences.

state S_0 . Layer 2 indicates the second layer, including states S_1, S_2, S_3, S_4 . Layer 3 indicates the third layer, including states $S_5, S_6, S_7, S_8, S_9, S_{10}, S_{11}, S_{12}$. Layer 4 indicates the fourth layer, including states $S_{13}, S_{14}, S_{15}, S_{16}, S_{17}, S_{18}, S_{19}, S_{20}, S_{21}, S_{22}$. Layer 5 indicates the fifth layer, including states $S_{23}, S_{24}, S_{25}, S_{26}, S_{27}, S_{28}, S_{29}, S_{30}, S_{31}$.

Layer 6 indicates the sixth layer, including states $S_{32}, S_{33}, S_{34}, S_{35}, S_{36}, S_{37}, S_{38}$. Layer 7 indicates the seventh layer, including states $S_{39}, S_{40}, S_{41}, S_{42}, S_{43}$. Layer 8 indicates the eighth layer, including states $S_{44}, S_{45}, S_{46}, S_{47}$. According to Fig. 10, there are seven transitions $a_{15}, a_{21}, a_{22}, a_{23}, b_{22}, b_{23}, c_{14}$ that can trigger to reach S_{47} . $\sigma_1 \cap \sigma_2 \cap \dots \cap \sigma_i \cap \sigma_j = \emptyset$; there is not only one path to S_{47} , and $\forall p \in P.M_{i-1}(p) \geq M_i(p)$.

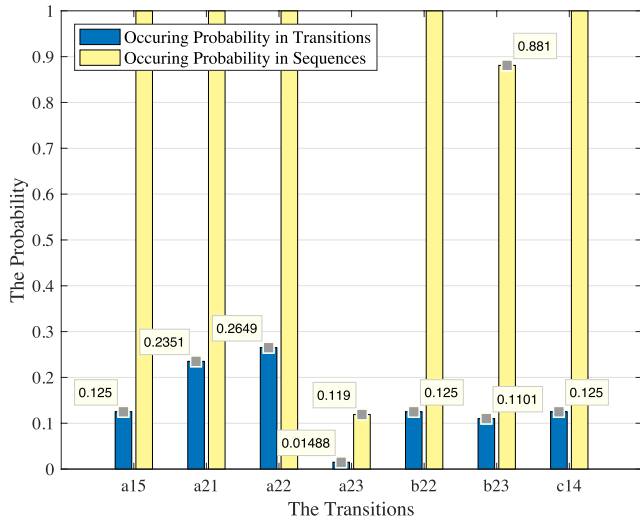


FIGURE 12. Firing probability of transitions in Fig. 8.

Then, according to Algorithm 1, $V_P = slice(M_{i-1}, \bigcup_{i=1}^n \sigma_i, P)$.

The vulnerable transitions set is as follows:

$$\begin{aligned}
 V_T &= \{t \mid p_i, p_i \in V_P\} \cup U_T \\
 &= \{a_{15}, a_{21}, a_{22}, a_{23}, b_{22}, b_{23}, c_{14}\} \cup \{b_{14}, b_{20}, b_{26}\} \\
 &= \{a_{15}, a_{21}, a_{22}, a_{23}, b_{22}, b_{23}, c_{14}, b_{14}, b_{26}\}. \quad (19)
 \end{aligned}$$

The normalized probabilities are given in Fig. 12. In Fig. 12, yellow shows the firing probability of transitions in each sequence, and blue shows the firing probability of transitions in all sequences. We can see that the probability of the transitions $a_{15}, a_{21}, a_{22}, b_{22}, b_{23}, c_{14}$ is 1 in each sequence. The probability of transitions a_{23} and b_{23} in each sequence is 0.1190 and 0.8810, respectively. The probabilities in all transitions are 0.125, 0.235119047619048, 0.264880952380952, 0.0148809523809524, 0.125, 0.110119047619048, and 0.125.

VI. CONCLUSION & FUTURE WORK

Due to the short times involved in online transactions and online payment platforms, online trading processes and trusted behavior issues are emerging along with the rapid development of online shopping and have gradually become a bottleneck in network trade development. Many e-commerce software systems are not mature and reliable, and they have flaws and mistakes that can be used by invaders. This leads to the emergence of security vulnerabilities and loss of user funds. This paper is motivated by the trusted behavior issues faced by these vulnerable network trade systems. Due to the uncomplicated graphical representation of a labeled Petri net, it can describe the overall structure of a local specification and unobservable actions well. However, the formal definition of its components can be used to provide precise abstraction. The VET-net is a subclass of labeled Petri nets. It can be used to model and simulate vulnerable e-commerce systems with unobservable actions. In this paper, on the basis of VET-nets,

we describe the concept of vulnerable transitions, which include not only vulnerable actions but also unobservable transitions. Based on the concept of a slice, we then present a new method to locate the vulnerable transitions of a VET-net. We use hidden Markov models (HMMs) to approximate the state reachability graph of a VET-net; this is called the HMM-RG method. Based on the HMM-RG, we describe the vulnerable state evaluation method of VET-nets. The vulnerable state evaluation method addresses the original problem. The proposed method is suitable for verifying and evaluating an online transaction system. It can also be used to verify simple e-commerce systems. The advantages of these methods are in dealing with unobservable actions. The proposed vulnerability assessment method can help designers analyze, diagnose and evaluate system vulnerabilities. Thus, the proposed method can be readily used in the system design and analysis of industrial online transaction business processes.

Due to neglecting data information, the methods are not suitable for some e-commerce systems, e.g., EBPN [4], [6], [13]. Regarding future work, there are other problems that require study related to the vulnerability evaluation of e-commerce systems, such as evaluation of data information and attack prevention.

ACKNOWLEDGMENT

The authors would like to thank the editors and the anonymous reviewers for their helpful comments.

REFERENCES

- [1] P. Zhao, Z. Ding, M. Wang, and R. Cao, "Behavior analysis for electronic commerce trading systems: A survey," *IEEE Access*, vol. 7, pp. 108703–108728, 2019.
- [2] Y. Du, C. Jiang, M. Zhou, and Y. Fu, "Modeling and monitoring of E-commerce workflows," *Inf. Sci.*, vol. 179, no. 7, pp. 995–1006, Mar. 2009.
- [3] Y. Du, L. Qi, and M. Zhou, "Analysis and application of logical Petri nets to E-commerce systems," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 44, no. 4, pp. 468–481, Apr. 2014.
- [4] W. Yu, C. Yan, Z. Ding, C. Jiang, and M. Zhou, "Modeling and validating E-commerce business process based on Petri nets," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 44, no. 3, pp. 327–341, Mar. 2014.
- [5] W. Yu, C. G. Yan, Z. Ding, C. Jiang, and M. Zhou, "Modeling and verification of online shopping business processes by considering malicious behavior patterns," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 647–662, Apr. 2016.
- [6] W. Yu, C. Yan, Z. Ding, C. Jiang, and M. Zhou, "Analyzing E-commerce business process nets via incidence matrix and reduction," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 1, pp. 130–141, Jan. 2018.
- [7] D. Xu and K. E. Nygard, "Threat-driven modeling and verification of secure software using aspect-oriented Petri nets," *IEEE Trans. Softw. Eng.*, vol. 32, no. 4, pp. 265–278, Apr. 2006.
- [8] Z. Ding, H. Qiu, R. Yang, C. Jiang, and M. Zhou, "Interactive-control-model for human-computer interactive system based on Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 4, pp. 1800–1813, Oct. 2019.
- [9] C. Jiang, Y. Fang, P. Zhao, and J. Panneerselvam, "Intelligent UAV identity authentication and safety supervision based on behavior modeling and prediction," *IEEE Trans. Ind. Informat.*, early access, Jan. 15, 2020, doi: 10.1109/THI.2020.2966758.
- [10] D. Chen, Z. Ding, C. Yan, and M. Wang, "A behavioral authentication method for mobile based on browsing behaviors," *IEEE/CAA J. Automatica Sinica*, early access, Jul. 18, 2019, doi: 10.1109/JAS.2019.1911648.
- [11] B. Yuan, J. Panneerselvam, L. Liu, N. Antonopoulos, and Y. Lu, "An inductive content-augmented network embedding model for edge artificial intelligence," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 4295–4305, Jul. 2019.

- [12] M. Wang, G. Liu, C. Yan, and C. Jiang, "Modeling and vulnerable points analysis for e-commerce transaction system with a known attack," in *Proc. Int. Conf. Secur., Privacy Anonymity Comput., Commun. Storage*, in LNCS. Berlin, Germany: Springer-Verlag, 2016, pp. 84–97.
- [13] M. Wang, Z. Ding, P. Zhao, W. Yu, and C. Jiang, "A dynamic data slice approach to the vulnerability analysis of e-commerce systems," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Aug. 21, 2018, doi: [10.1109/TSMC.2018.2862387](https://doi.org/10.1109/TSMC.2018.2862387).
- [14] D. You, S. Wang, and C. Seatzu, "Verification of fault-predictability in labeled Petri nets using predictor graphs," *IEEE Trans. Autom. Control*, vol. 64, no. 10, pp. 4353–4360, Oct. 2019.
- [15] B. O. Emeka and S. Liu, "Assessing and extracting software security vulnerabilities in SOFL formal specifications," in *Proc. Int. Conf. Electron., Inf., Commun. (ICEIC)*, Jan. 2018, pp. 1–4.
- [16] X. Fang, C. Jiang, Z. Yin, and X. Fan, "The trustworthiness analyzing of interacting business process based on the induction information," *Comput. Sci. Inf. Syst.*, vol. 8, no. 3, pp. 843–867, 2011.
- [17] L. Allodi and F. Massacci, "Security events and vulnerability data for cybersecurity risk estimation," *Risk Anal.*, vol. 37, no. 8, pp. 1606–1627, Aug. 2017.
- [18] X. Wang, C. Mahulea, and M. Silva, "Diagnosis of time Petri nets using fault diagnosis graph," *IEEE Trans. Autom. Control*, vol. 60, no. 9, pp. 2321–2335, Sep. 2015.
- [19] L. Li and C. N. Hadjicostis, "Minimum initial marking estimation in labeled Petri nets," *IEEE Trans. Autom. Control*, vol. 58, no. 1, pp. 198–203, Jan. 2013.
- [20] O. Prakash, A. K. Samantaray, and R. Bhattacharyya, "Model-based diagnosis of multiple faults in hybrid dynamical systems with dynamically updated parameters," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 6, pp. 1053–1072, Jun. 2019.
- [21] D. Lefebvre, "On-line fault diagnosis with partially observed Petri nets," *IEEE Trans. Autom. Control*, vol. 59, no. 7, pp. 1919–1924, Jul. 2014.
- [22] R. M. Al-Dwairi and M. A. Kamala, "Business-to-consumer e-commerce Web sites: Vulnerabilities, threats and quality evaluation model," in *Proc. 20th Int. Conf. Electron. Commun. Comput. (CONIELECOMP)*, Feb. 2010, pp. 206–211.
- [23] H. Wang, Z. Chen, J. Zhao, X. Di, and D. Liu, "A vulnerability assessment method in industrial Internet of Things based on attack graph and maximum flow," *IEEE Access*, vol. 6, pp. 8599–8609, 2018.
- [24] J. Fonseca, M. Vieira, and H. Madeira, "Evaluation of Web security mechanisms using vulnerability & attack injection," *IEEE Trans. Dependable Secure Comput.*, vol. 11, no. 5, pp. 440–453, Sep. 2014.
- [25] N. Pedroni, *Advanced Methods for the Risk, Vulnerability and Resilience Assessment of Safety-Critical Engineering Components, Systems and Infrastructures, in the Presence of Uncertainties*. Bengaluru, India: HAL, 2017.
- [26] A. Khalid, O. Lee, M. Choi, and J. Ahn, "The effects of customer satisfaction with E-commerce system," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 2, pp. 481–491, 2018.
- [27] I. Najafi, "E-trust building models, methods and enhancing on B2C E-commerce companies," *Int. J. Comput. Inf. Technol.*, vol. 4, pp. 119–128, 2015.
- [28] D. A. Grejner-Brzezinska, C. K. Toth, T. Moore, J. F. Raquet, M. M. Miller, and A. Kealy, "Multisensor navigation systems: A remedy for GNSS vulnerabilities?" *Proc. IEEE*, vol. 104, no. 6, pp. 1339–1353, Jun. 2016.
- [29] M. P. Cabasino, A. Giua, and C. Seatzu, "Diagnosis using labeled Petri nets with silent or undistinguishable fault events," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 43, no. 2, pp. 345–355, Mar. 2013.
- [30] Y. Hu, Z. Ma, and Z. Li, "Design of supervisors for active diagnosis in discrete event systems," *IEEE Trans. Autom. Control*, early access, Jan. 28, 2020, doi: [10.1109/TAC.2020.2970011](https://doi.org/10.1109/TAC.2020.2970011).
- [31] F. Basile, M. P. Cabasino, and C. Seatzu, "State estimation and fault diagnosis of labeled time Petri net systems with unobservable transitions," *IEEE Trans. Autom. Control*, vol. 60, no. 4, pp. 997–1009, Apr. 2015.
- [32] Y. Shoukry, P. Nuzzo, A. Puggelli, A. L. Sangiovanni-Vincentelli, S. A. Seshia, and P. Tabuada, "Secure state estimation for cyber-physical systems under sensor attacks: A satisfiability modulo theory approach," *IEEE Trans. Autom. Control*, vol. 62, no. 10, pp. 4917–4932, Oct. 2017.
- [33] P. Bonhomme, "Marking estimation of P-time Petri nets with unobservable transitions," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 45, no. 3, pp. 508–518, Mar. 2015.
- [34] S. Koga, M. Diagne, and M. Krstic, "Control and state estimation of the one-phase Stefan problem via backstepping design," *IEEE Trans. Autom. Control*, vol. 64, no. 2, pp. 510–525, Feb. 2019.
- [35] X. Hu, W. Zhang, X. Hu, N. Yu, X. Zhao, and F. Li, "Fast estimation of optimal marked-signal distribution for reversible data hiding," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 5, pp. 779–788, May 2013.
- [36] Z. Yu, L. Zhou, Z. Ma, and M. A. El-Meligy, "Trustworthiness modeling and analysis of cyber-physical manufacturing systems," *IEEE Access*, vol. 5, pp. 26076–26085, 2017.
- [37] Y. Li, L. Yin, Y. Chen, Z. Yu, and N. Wu, "Optimal Petri net supervisor synthesis for forbidden state problems using marking mask," *Inf. Sci.*, vol. 505, pp. 183–197, Sep. 2019.
- [38] X. Zan, Z. Wu, C. Guo, and Z. Yu, "A Pareto-based genetic algorithm for multi-objective scheduling of automated manufacturing systems," *Adv. Mech. Eng.*, vol. 12, no. 1, Jan. 2020, Art. no. 168781401988529.
- [39] M. Zhou and K. Venkatesh, *Modeling, Simulation, and Control of Flexible Manufacturing Systems: A Petri Net Approach*. Singapore: World Scientific, 1998.
- [40] M. Wang, Z. Ding, G. Liu, C. Jiang, and M. Zhou, "Measurement and computation of profile similarity of workflow nets based on behavioral relation matrix," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Jul. 24, 2018, doi: [10.1109/TSMC.2018.2852652](https://doi.org/10.1109/TSMC.2018.2852652).
- [41] M. Wang, G. Liu, P. Zhao, C. Yan, and C. Jiang, "Behavior consistency computation for workflow nets with unknown correspondence," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 1, pp. 281–291, Jan. 2018.
- [42] S. Wang, D. You, M. Zhou, and C. Seatzu, "Characterization of admissible marking sets in Petri nets with uncontrollable transitions," *IEEE Trans. Autom. Control*, vol. 61, no. 7, pp. 1953–1958, Jul. 2016.
- [43] Z. Ding, M. Pan, Y. Ru, C. Jiang, and M. Zhou, "Fully expanded tree for property analysis of one-place-unbounded Petri nets," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 9, pp. 2574–2585, Sep. 2017.
- [44] Z. Ding, Y. Zhou, and M. Zhou, "A polynomial algorithm to performance analysis of concurrent systems via Petri nets and ordinary differential equations," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 1, pp. 295–308, Jan. 2015.
- [45] X. Fang, C. Jiang, and X. Fan, "Independent global constraints-aware Web service composition optimization," *Inf. Technol. J.*, vol. 8, no. 2, pp. 181–187, Feb. 2009.
- [46] X. Fang, W. Hao, X. Fan, and Z. Yin, "The analysis method about change domain of business process model based on the behavior profile of Petri net," *Appl. Math. Inf. Sci.*, vol. 6, pp. 943–949, Nov. 2012.
- [47] X. Liu, M. Wang, and L. Liu, "The analysis of change region about networked control system based on the behavior profile," *Adv. Intell. Syst. Comput.*, vol. 212, no. 2013, pp. 1221–1228, 2013.
- [48] J.-M. Proth and X.-L. Xie, "Cycle time of stochastic event graphs: Evaluation and marking optimization," *IEEE Trans. Autom. Control*, vol. 39, no. 7, pp. 1482–1486, Jul. 1994.
- [49] C. R. Vazquez and M. Silva, "Stochastic hybrid approximations of Markovian Petri nets," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 45, no. 9, pp. 1231–1244, Sep. 2015.
- [50] F. G. Cabral, M. V. Moreira, O. Diene, and J. C. Basilio, "A Petri net Diagnoser for discrete event systems modeled by finite state automata," *IEEE Trans. Autom. Control*, vol. 60, no. 1, pp. 59–71, Jan. 2015.
- [51] V. Kumar, L. K. Singh, P. Singh, K. V. Singh, A. K. Maurya, and A. K. Tripathi, "Parameter estimation for quantitative dependability analysis of safety-critical and control systems of NPP," *IEEE Trans. Nucl. Sci.*, vol. 65, no. 5, pp. 1080–1090, May 2018.
- [52] P. Declerck and P. Bonhomme, "State estimation of timed labeled Petri nets with unobservable transitions," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 1, pp. 103–109, Jan. 2014.
- [53] M. P. Cabasino, A. Giua, and C. Seatzu, "Diagnosability of discrete-event systems using labeled Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 1, pp. 144–153, Jan. 2014.
- [54] Y. Tong, Z. Li, and A. Giua, "On the equivalence of observation structures for Petri net generators," *IEEE Trans. Autom. Control*, vol. 61, no. 9, pp. 2448–2462, Sep. 2016.
- [55] F. Basile, M. P. Cabasino, and C. Seatzu, "Diagnosability analysis of labeled time Petri net systems," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1384–1396, Mar. 2017.
- [56] Y. Ephraim and B. L. Mark, "Causal recursive parameter estimation for discrete-time hidden bivariate Markov chains," *IEEE Trans. Signal Process.*, vol. 63, no. 8, pp. 2108–2117, Apr. 2015.
- [57] C. Lee, H. Shim, and Y. Eun, "On redundant observability: From security index to attack detection and resilient state estimation," *IEEE Trans. Autom. Control*, vol. 64, no. 2, pp. 775–782, Feb. 2019.

- [58] Y. Liu, F. Han, J. Wang, and H. Qi, "Vulnerability assessment of a multistate component for IEMI based on a Bayesian method," *IEEE Trans. Electromagn. Compat.*, vol. 61, no. 2, pp. 467–475, Apr. 2019.
- [59] S. Pinchinat, M. Acher, and D. Vojtisek, "Towards synthesis of attack trees for supporting computer-aided risk analysis," in *Proc. Int. Conf. Softw. Eng. Formal Methods*. Cham, Switzerland: Springer, 2014, pp. 363–375.
- [60] S. Mauw and M. Oostdijk, "Foundations of attack trees," in *Proc. Int. Conf. Inf. Secur. Cryptol.* Berlin, Germany: Springer, 2005, pp. 186–198.
- [61] K. Mikko, T. Venäläinen, and S. Kinnunen, "Towards modelling information security with key-challenge Petri nets," in *Proc. Nordic Conf. Secure IT Syst.*, in LNCS, vol. 5838. Berlin, Germany: Springer-Verlag, 2009, pp. 190–206.
- [62] B. Kordy, S. Mauw, S. Radomirović, and P. Schweitzer, "Foundations of attack–defense trees," in *Proc. Int. Workshop Formal Aspects Secur. Trust*, in LNCS, vol. 6561. Berlin, Germany: Springer-Verlag, 2010, pp. 80–95.
- [63] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden Markov model," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1992, pp. 379–385.
- [64] A. Rakow, "Slicing Petri nets with an application to workflow verification," in *Proc. Int. Conf. Current Trends Theory Pract. Comput. Sci.* Berlin, Germany: Springer, 2008, pp. 436–447.
- [65] M. Llorens, J. Oliver, J. Silva, S. Tamarit, and G. Vidal, "Dynamic slicing techniques for Petri nets," *Electron. Notes Theor. Comput. Sci.*, vol. 223, pp. 153–165, Dec. 2008.
- [66] M. P. Cabasino, C. N. Hadjicostis, and C. Seatzu, "Probabilistic marking estimation in labeled Petri nets," *IEEE Trans. Autom. Control*, vol. 60, no. 2, pp. 528–533, Feb. 2015.



MIMI WANG (Member, IEEE) received the M.S. degree in mathematics and applied mathematics from the Anhui University of Science and Technology, Huainan, China, 2013, and the Ph.D. degree in computer science and technology from Tongji University, Shanghai, China, in 2019. She is currently a Lecturer with Department of Automation, College of Information Science and Technology, Donghua University. She has authored over 30 technical articles in journals and conference proceedings. Her research interests include Petri nets, diagnosability analysis, optimization, formal verification of software, and security in e-commerce systems. She is also a very active Reviewer for many international journals.



ZHIJUN DING (Senior Member, IEEE) received the M.S. degree from the Shandong University of Science and Technology, Qingdao, China, in 2001, and the Ph.D. degree from Tongji University, Shanghai, China, in 2007.

He is currently a Professor with the Department of Computer Science and Technology, Tongji University. He has authored over 100 articles in journals and conference proceedings. His current research interests include formal engineering, Petri nets, services computing, and workflows.



PEIHAI ZHAO (Member, IEEE) received the B.S. and Ph.D. degrees from Tongji University, Shanghai, China, in 2011 and 2017, respectively.

He is currently a Lecturer with the School of Computer Science and Technology, Donghua University. He has authored over 20 technical articles in journals and conference proceedings, including *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, the *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS*, the *IEEE/CAA JOURNAL OF AUTOMATICA SINICA*, and *IEEE ACCESS*. He holds over ten patents, including two international patents. His current research interests include Petri nets, process mining, discrete event systems, electronic commerce systems, formal verification of software, optimization algorithms, data mining, artificial intelligence, and machine learning.

• • •