Chaotic Activity Filtering Approach of Business Process Based on The Directly-Follows Relation

Tengzi Lv [1], Xiugang Gong [2*] ,Muhammad Tahir[3], Na Gong[4], Kaiyu Li[5]
[2]Corresponding Author: gong_xg@sdut.edu.cn 1154835115@qq.com
[1,2*,4,5]School of Computer Science and Technology, Shandong University of Technology, Zibo 255049 China
[3]Department of Computer Science, Mohammad Ali Jinnah University, Block 6, P.E.C.H.S, Karachi, 75400, Pakistan

Abstract
Process discovery aims to discover process models from event logs to describe actual business processes. However, chaotic activities may exist in real business scenarios, and the occurrence of chaotic activities is independent of other activities in the process and can occur at any location in the event log at any frequency. Therefore, chaotic activities seriously affect the approach quality of process discovery. Filtering chaotic activities in event logs can effectively improve the quality of event logs and thus improve the quality of process models. The traditional chaotic activity filtering algorithm is difficult to balance accuracy and time performance. Therefore, new approaches for filtering chaotic activities are proposed in this paper. By analyzing the relation between activities, chaotic activities are identified in the log according to the characteristics of chaotic activities and the directly-follows relation of activities as the judgment conditions, and the filtering of chaotic activities in the event log is realized. The proposed approaches are compared with the traditional chaotic activity filtering approaches on several simulation/real data sets, and the accuracy and running time between the multi-group event logs and the process models generated before and after chaotic activity filtering are analyzed, further verifying the effectiveness and feasibility of the proposed approaches.

Keywords:
process mining; chaotic activities; process model; directly-follows relation; event log.

1. Introduction

Process mining realizes the integration of various business links by modeling, managing, monitoring and optimizing the whole life cycle of business processes. The research on process mining focuses on the analysis of event data recorded during the execution of business processes. Process discovery is the basic research direction in process mining. Process discovery technology aims to generate business process model through the information contained in event logs. However, in the actual process execution, there are some activities that occur spontaneously at any point in time and are not part of the process, such activities are chaotic activities. The existence of chaotic activities greatly affects the quality of process models obtained by process discovery techniques, and the occurrence of chaotic activities will not change the probability of other activities. Therefore, the process of discovering the process model from the event log containing chaotic activities is very complicated, and the existence of chaotic activities will lead to excessive generalization of the obtained model.

In the actual scene, in order to get a high-quality process model, it is necessary to preprocess the event log, that is, to find out the activities, events or tracks that affect the quality of the process model and filter them from the log to improve the quality of the model. Similar event log preprocessing method also includes filtering noise and low-frequency activities. However, the method of filtering noise or infrequent activities in the event log has some shortcomings in solving the problem of chaotic activities in the log. It has been proposed in some literatures that chaotic activities can be identified and filtered by calculating activity entropy. However, due to the complexity of entropy calculation, it takes a lot of time to filter chaotic activities with event logs containing many tracks and activities, and the efficiency of this method is low in the filtering of chaotic activities.

In this paper, new chaotic activity filtering approaches based on the directly-follows relation between activities are proposed, and chaotic activities are identified by using the algorithm to count the directly-follows relation between each activity and other activities. Firstly, the directly-follows relation between activities in the log is calculated, and then the

Manuscript received February 5, 2025
Manuscript revised February 20, 2025
https://doi.org/10.22937/IJCSNS.2025.25.2.18

chaos degree of each activity in the log and the influence of the existence of activities in the log on the total chaos degree of the log are obtained. Finally, the obtained results are used to set conditions to judge whether the activity is chaotic. Experiments show that the approaches can give consideration to both accuracy and running time.

The first section of this paper introduces related work, the second section introduces two filtering approaches based on directly-follows relation, the third section introduces the comparative experiment between the proposed method and the traditional method, and the fourth section summarizes the article.

2. Related Work

In process mining, most of the existing work of filtering event logs adopts the following methods: first identify the behavior in the event log that affects the quality of the model obtained through process mining, and then delete the behavior from the log, thus improving the quality of process model. The existing filtering approaches in the field of process mining can be divided into four categories: event filtering technology, process discovery technology with built-in filtering mechanism, trace filtering technology and activity filtering technology.

Event filtering technology. Event filtering technology aims to filter out outliers from event logs while maintaining mainstream behavior. Confirti et al. [1] proposed to use integer linear programming solver to construct prefix automata of event logs, and to remove infrequent arcs from the minimal prefix automata. Lu et al. [2] advocate the use of event mapping to distinguish between events and abnormal events as part of the mainstream behavior of the process. Fani Sani et al. [3] proposed to use sequential pattern mining technology to distinguish events belonging to mainstream behavior from abnormal events.

Process discovery technology with built-in filtering mechanism. Some process discovery techniques can delete infrequent elements when mining models to get high-quality models. In addition to directly-follows relation, Heuristic Miner [4] and the Fodina algorithm [5] also defines the eventually-follows relation between activities and filters out unusual directly-follows relation and eventually-follows relation. The eventually-follows relation is different from the directly-follows relation and is not affected by chaotic activities. Inductive Miner [6] is a process discovery algorithm, which first finds a directly- follows graph from the event log, and then discovers the process model in the second step. Inductive Miner in frequency (IMf) [7] is an extension of Inductive Miner, and this method filters out uncommon direct follow relations from the set of direct follow relations used to generate models.

Trace filtering technology. The purpose of trace filtering technology is to identify and delete the traces that affect the quality of the model in the event log. Ghionna et al. [8] proposed mining frequent patterns from event logs and applying MCL clustering to traces. Traces that are not assigned to clusters by MCL clustering algorithm are regarded as outlier traces and filtered from event logs. Cheng and Kumar [9] proposed a supervised approach to filter out noisy traces from event logs, assuming that there is a sub-log that has been manually checked and marked. PRISM rule induction algorithm is used to extract classification rules, and clean and noisy tracks are distinguished by training on labeled sub-logs, and then these classification rules are applied to identify and filter noisy tracks from unlabeled sub-logs.

Activity filtering technology. The activity filtering technology is to filter out infrequent activities from the event log. Yi Guo et al. [10] propose a chaotic activity filtering method based on bidirectional causal dependency. By analyzing the bidirectional causal dependency between the model and the event log, the accuracy between the model and the event log is used as a constraint to achieve filtering of chaotic activities in the event log. The Inductive Visual Miner [11] proposed by Leemans et al. is an interactive process discovery tool, which implements the Inductive Miner process discovery algorithm in an interactive way: process analysts can use sliders to filter event logs. Tax N proposed a new technology to filter chaotic activities from event log, which calculated the entropy of the activity to judge whether the activity was chaotic, and used direct or indirect approach to filter chaotic activities from the event log.

Most of the existing chaotic activity filtering work is to identify chaotic activities by deleting noise or based on the occurrence frequency of activities, and filter them. However, these approaches may lead to more deletion or missing deletion of activities, which can not solve the problem

of chaotic activities affecting the quality of event logs. However, the chaotic activity filtering approaches based on activity entropy has high time complexity and low efficiency when filtering chaotic activities. Therefore, this study proposes chaotic activity filtering approaches based on the directly-follows relation between activities, which filters chaotic activities in the event log by counting the relation between each activity and other activities. This method can reduce the time required for filtering chaotic activities and effectively improve the quality of logs while ensuring the accuracy.

3. Chaotic Activity Filtering Approach Based on Directly-follows Relation

This section introduces two methods of filtering chaotic activities in the log: direct chaotic activity filtering and indirect chaotic activity filtering, through which chaotic activities in the event log can be identified and deleted.

3.1. Chaotic Activity Filtering Approach Based on Directly-follows Relation

The traditional framework of chaotic activity filtering technology is divided into three stages. In the first stage, chaotic activity is identified from the original log by chaotic activity identification technology. In the second stage, the identified chaotic activities are deleted from the log to get the filtered event log. In the third stage, the process model is obtained from the new log obtained after filtering chaotic activities by using the process discovery algorithm. This paper proposes a method for filtering chaotic activities in business processes based on the directly-follows relation between activities. As shown in **Fig. 1**, firstly, the directly-follows relation between activities in the event log and the set of activities contained in the log are extracted, then which activities in the log are chaotic according to the directly-follows relation between activities in the event log and other activities, and finally the identified chaotic activities are deleted from the set of activities in the original log. The delete method converts the new activity collection into the new event log by deleting activities that are not included in the new activity collection from the original log.

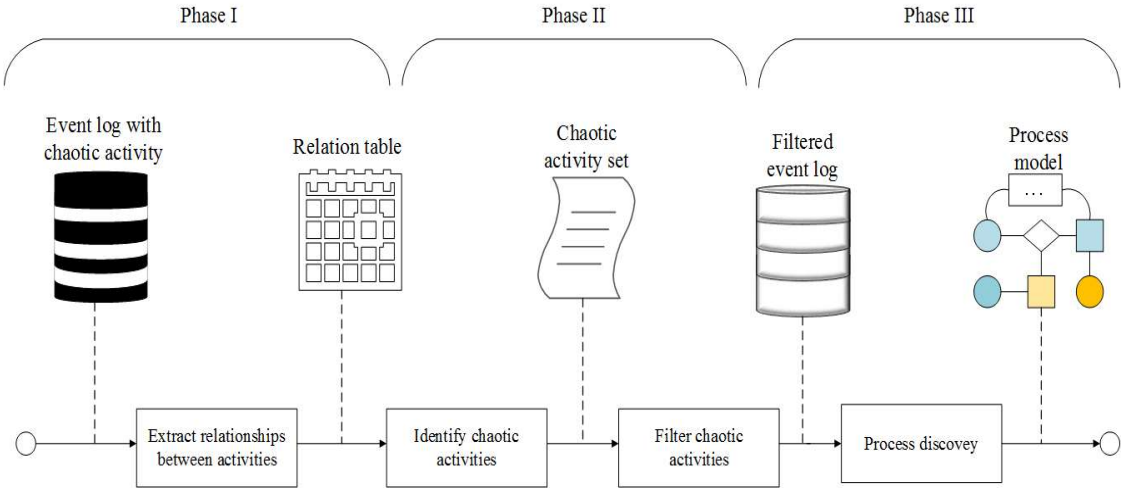


Fig.1. Chaotic activity filtering based on the directly-follows relation.

Define 1 (event, trace, event log) and let E be the event universe, that is, the set of all possible event identifiers. Events can be characterized by various attributes. Trace is a finite sequence of events $\sigma \in E^*$, so that each event only appears once in the trace, i.e., for $1 \le i \le j \le |\sigma| : \sigma(i) \ne \sigma(j)$. The event log is a set of case $L \subseteq C$, so that each event appears at most once in the entire log, i.e., for any $c_1, c_2 \in L, c_1 \ne c_2 : \partial_{set}(\#trace(c_1)) \cap \partial_{set}(\#trace(c_2)) = \varnothing$.

Activity	a		b		c		x	
	d_1	d_2	d_1	d_2	d_1	d_2	d_1	d_2
a	0	0	20	0	0	0	10	0
b	0	20	0	0	20	0	10	10
c	0	0	0	20	0	0	10	10
x	0	10	10	10	10	10	0	0

L=[$\langle a,b,c\rangle^2$,$\langle b,a,c\rangle^3$] is an example event log of process activity set $\{a,b,c\}$, which is composed of two traces $\langle a,b,c\rangle$ and three traces $\langle b,a,c\rangle$. *Activities*(L) represents the set of process activities that occur in L, for example, *Activities*(L)= $\{a,b,c\}$. #(a,L) represents the number of occurrence that activity a appears in log L, for example, #(a,L)=5.

Extend the function #(a,L) to #(σ,L) to count the number of occurrence a sequence appears in the log.

$$\#(\sigma, L) = \sum_{\sigma' \in L} \left| \left\{ 0 \le i \le |\sigma'| - |\sigma| \middle| \forall_{1 \le j \le |\sigma|} \sigma'(i+j) = \sigma(i) \right\} \right| \quad (1)$$

Definition 2 (multiple sets) $X = \{a_1,a_2,...,a_n\}$ denotes a finite set. $X \backslash Y$ represents a set of elements in set X but not in set Y, for example, $\{a,b,c\} \backslash \{a,c\} = \{b\}$. X^* represents the set of all finite sequences on set X. $\sigma = \langle a_1,a_2,...,a_n \rangle$ denotes a sequence of length n, where $\sigma(i) = a_i$ and $\langle\rangle$ is an empty sequence. $\sigma \uparrow X$ is the projection of σ on X, such as $\langle a,b,c,a,b,c\rangle \uparrow \{a,c\} = \langle a,c,a,c\rangle$. $\sigma_1 \cdot \sigma_2$ denotes the concatenation of σ_1 and σ_2, for example, $\langle a,b,c\rangle \cdot \langle d,e\rangle = \langle a,b,c,d,e\rangle$.

Definition 3 (relation table) Let D be a relation table, D_{ij} be the element in the i row and the j column in table D, i and j be the activities contained in the log, and let d_1 = #(i,j) be the directly-follows frequency between activities i and j, that is, the number of occurrence that j directly follows i, and d_2 = #(j,i) be the directly-precedes frequency between activities i and j, that is, the number of occurrence that i directly follows j. Take the event log L=[$\langle a,b,c,x\rangle^{10}$,$\langle a,b,x,c\rangle^{10}$,$\langle a,x,b,c\rangle^{10}$] as an example. **Table I** shows the relation between activities in the log.

TABLE I. The relation table of L=[$\langle a,b,c,x\rangle^{10}$,$\langle a,b,x,c\rangle^{10}$,$\langle a,x,b,c\rangle^{10}$].

3.2. Direct Chaotic Activity Filtering

Algorithm 1 describes an algorithm for iteratively filtering chaotic activity from event logs. The algorithm takes the event log L as input and generates an event log list. Each element in the list contains a filtered version of L, and compared with the previous element, each subsequent element of the list has additional activities filtered out. The process of the algorithm proposed in this section is as follows: firstly, the relation between activities in the log is extracted from the input event log to form a relation table, and then the relation table is used to calculate the statistics of the relation between activities to determine which activities are chaotic activities. Generally speaking, chaotic activities have no clear position in the log, and the directly-follows relation between chaotic activities and other activities is in disorder. Therefore, in the event log, the directly-follows relation and directly-precedes relation between chaotic activities and other activities are confusing, i.e., chaotic activities and most activities in the log have different follows relations. Therefore, we can get the corresponding activity set that meets the following conditions by counting the different follows relations between activities, and calculate the chaos degree of the activities through the obtained activity set, and judge whether the activities are chaotic according to the chaos degree of the activities. Define the chaos degree of activity $y \in Activities(L)$ in the log for the *i-th* condition as $CH_i(y)$.

Condition (1): The sum of the number of directly-follows relation and the number of directly-precedes relation that exist between an activity and all other activities in the log was used to represent the degree of activity disorder, i.e., for activity x, let the event log be L, the sum of the number of

elements in the set $\left\{y \middle| \forall_{y \in Activities(L)} \#(< x,y >,L) > 0\right\}$ and $\left\{y \middle| \forall_{y \in Activities(L)} \#(< y,x >,L) > 0\right\}$ be counted as the degree of chaos in activity x. If the chaotic degree of activity x exceeds the set threshold, it is judged that x is a candidate chaotic activity. For example, for log L=[<a,b,c,x>^{10},<a,b,x,c>^{10},<a,x,b,c>^{10}], for activity x, the set of activities that meet the conditions is {b,c} and {a,b,c}, and calculate the number of elements in the resulting set, so $CH_1(x)$=5 is obtained.

Condition (2): Chaotic activities usually have both directly-follows relation and directly-precedes relation with more activities in the log. Therefore, the number of activities in the event log that have both directly-follows relation and directly-precedes relation with a certain activity can be counted to calculate the chaos degree of the activity. i.e., for activity x, let the event log be L, the sum of the number of elements in the set $\left\{y \middle| \forall_{y \in Activities(L)} \left(\#(< x,y >,L) > 0\right) \wedge \left(\#(< y,x >,L) > 0\right)\right\}$ be counted as the degree of chaos in activity x. If the chaos degree of activity x exceeds the set threshold, it is judged that x is a candidate chaotic activity. For example, for log L=[<a,b,c,x>^{10},<a,b,x,c>^{10},<a,x,b,c>^{10}], for activity x, the set of activities that meet the conditions is {b,c}, and calculate the number of elements in the resulting set, so $CH_2(x)$=2 is obtained.

Condition (3): If there is both directly-follows relation and directly-precedes relation between a certain activity and more activities, and the difference between the directly-follows frequency and the directly-precedes frequency is small, the activity is judged to be chaotic. The average value of the directly-follows frequency and the directly-precedes frequency between two activities can be set as the threshold to judge the gap between them. Therefore, activities that have both directly-follows and directly-precedes relation with a certain activity can first be counted in the log to form an activity set. Then, in this set, the activities with a small difference between the directly-follows frequency and the directly-precedes frequency between the activity are counted to form the activity set. Finally, the number of activities in the activity set can be counted to calculate the chaos degree of the activity. i.e., for activity x, let the event log be L, the sum of the number of elements in the set $\left\{\begin{array}{l} y \middle| \forall_{y \in Activities(L)} \left(\#(< x,y >,L) > 0\right) \wedge \left(\#(< y,x >,L) > 0\right) \wedge \\ \left(\left|\#(< x,y >,L) - \#(< y,x >,L)\right| < \dfrac{\#(< x,y >,L) + \#(< y,x >,L)}{2}\right) \end{array}\right\}$ be counted as the degree of chaos in activity x. If the chaos degree of activity x exceeds the set threshold, it is judged that x is a candidate chaotic activity. for log L=[<a,b,c,x>^{10},<a,b,x,c>^{10},<a,x,b,c>^{10}], for activity x, the set of activities that meet the conditions is {b,c}, and calculate the number of elements in the resulting set, so $CH_3(x)$=2 is obtained.

Condition (4): By dividing the value obtained from condition (3) with the value obtained from condition (2), the set of activities that have both direct following relationship and direct preceding relationship with an activity in the log is obtained through condition (2), and through condition (3), we get the set of activities with small difference between the directly-follows frequency and the directly-precedes frequency between the activity in the log. Calculate the proportion of the latter to the former, which is the chaos degree of the activity . If the value calculated for an activity is 0 for condition (2), the value calculated for that activity is 0 for condition (4). A threshold is set, and if the chaos degree of activity x exceeds this threshold, it is judged that x is a candidate chaotic activity. For example, for log L=[<a,b,c,x>^{10},<a,b,x,c>^{10},<a,x,b,c>^{10}], for activity x, $CH_2(x)$=2, $CH_3(x)$=2, so $CH_4(x)$ is obtained.

According to this idea, the criterion for judging chaotic activities is set as:

Standard (1): Calculate the sum of the number of occurrence that each activity has directly-follows relation with all other activities in the log and the number of occurrence that each activity has directly-precedes relation with all other activities in the log, so as to judge whether the result calculated for an activity is greater than the average value of the sum of the results calculated for all activities in turn. If so, the activity is a candidate chaotic activity.

Take the log L=[<a,b,c,x>^{10},<a,b,x,c>^{10},<a,x,b,c>^{10}] as an example, and Table 1 is used to calculate for activity x. First of all, for activity x, find the

activities that have directly-follows relation with x in the relation table, and get that the values of $\#(<x,b>,L)$ and $\#(<x,c>,L)$ are all greater than 0, so the set of activities that have directly-follows relation with x is $\{b,c\}$, and the number of elements in the set is 2. Secondly, for activity x, find the activities that have a directly-precedes relation with x in the relation table, and get that the values of $\#(<b,x>,L)$, $\#(<b,x>,L)$ and $\#(<c,x>,L)$ are all greater than 0, so the set of activities that have a directly-follows relation with x is $\{a,b,c\}$, and the number of elements in the set is 3. The sum of the number of elements in two sets is calculated, and the result is 5. This calculation is performed for all activities in the log, and the corresponding results of each activity are obtained, and the average value of all results is obtained. In this log, the average value is 3.5. For activity b and x, the calculated result is greater than the average value, so it is judged that the candidate chaotic activities are b and x.

Standard (2): Count the number of occurrence that each activity has both directly-follows relation and directly-precedes relation with all other activities in the log, so as to judge whether the result calculated for an activity is greater than the average value of the sum of the results calculated for all activities in turn. If so, the activity is a candidate chaotic activity. so as to judge whether the result calculated for an activity is greater than the average value of the sum of the results calculated for all activities in turn. If so, the activity is a candidate chaotic activity.

Take the log $L=[<a,b,c,x>^{10},<a,b,x,c>^{10},<a,x,b,c>^{10}]$ as an example, and use Table 1 to calculate activity x.. First of all, for activity x, find the activities that have both directly-follows relation and directly-precedes relation with x in the relation table, and get that $\#(<x,b>,L)$, $\#(<b,x>,L)$ are all greater than 0, and the values of $\#(<x,c>,L)$, $\#(<c,x>,L)$ are all greater than 0. This calculation is performed for all activities in the log, and the corresponding results of each activity are obtained, and the average value of all results is obtained. In this log, the average value is 1. The result of activity x is greater than the average, so it is judged that the candidate chaotic activity is x.

Standard (3): For each activity, count the number of times that both the direct following frequency and the direct preceding frequency exist between the activity and all other activities in the log and the difference between the direct following frequency and the direct preceding frequency is not zero and does not exceed the average sum of the two, In this way, it is judged whether the calculated result of an activity is greater than the average value of the sum of the calculated results of all activities in turn. If so, the activity is a candidate chaotic activity.

Take the log $L=[<a,b,c,x>^{10},<a,b,x,c>^{10},<a,x,b,c>^{10}]$ as an example, and use Table 1 to calculate activity x. First of all, for activity x, find the activities that have both directly-follows relation and directly-precedes relation with x in the relation table, and get that $\#(<x,b>,L)$, $\#(<b,x>,L)$ are all greater than 0 and $\left|\#(<x,b>,L)-\#(<b,x>,L)\right|<\frac{\#(<x,b>,L)+\#(<b,x>,L)}{2}$, $\#(<x,c>,L)$, $\#(<c,x>,L)$ greater than 0 and $\left|\#(<x,c>,L)-\#(<c,x>,L)\right|<\frac{\#(<x,c>,L)+\#(<c,x>,L)}{2}$. This calculation is performed for all activities in the log, and the corresponding results of each activity are obtained, and the average value of all results is obtained. In this log, the average value is 1. The result of activity x is greater than the average, so it is judged that the candidate chaotic activity is x.

Standard (4): First, the calculation is made for condition (2),(3), and divide the calculation result obtained by the condition (3) by the calculation result obtained by the condition (2) to get the result. Then, judge whether the calculation result for an activity is greater than the average value of the sum of the results calculated for all activities in turn. If so, the activity is a candidate chaotic activity.

Take the log $L=[<a,b,c,x>^{10},<a,b,x,c>^{10},<a,x,b,c>^{10}]$ as an example, and use Table 1 to calculate activity x. Firstly, for activity x in the event log, the condition (2) is calculated, and $CH_2(x)=2$ is obtained; For condition (3), $CH_3(x)=2$, so for condition (4), the calculated result is $CH_4(x)=1$. This calculation is performed for all the activities in the log, and the corresponding results of each activity are obtained. For all the results, the average value is 0.75. The results of activities b, c and x are greater than the average, so it is judged that the candidate chaotic activities are b, c and x.

Take the event log $L=[<a,b,c,x>^{10},<a,b,x,c>^{10},<a,x,b,c>^{10}]$ as an example. For the condition(1), we obtain $\{CH_1(a)=2,\ CH_1(b)=4,\ CH_1(c)=3,\ CH_1(x)=5\}$, with an average of 3.5, so the candidate chaotic activities are b, x. For the condition (2), it is obtained that $\{CH_2(a)=0, CH_2(b)=1, CH_2(c)=1, CH_2(x)=2\}$, the average value is 1, so the candidate chaotic activity is x. For condition (3), we obtain that $\{CH_3(a)=0, CH_3(b)=1, CH_3(c)=1, CH_3(x)=2\}$, the average value is 1, so the candidate chaotic activity is x. For condition (4), it is obtained that $CH_4(a)=0$, $CH_4(b)=1$, $CH_4(c)=1$, $CH_4(x)=1\}$, with an average value of 0.75, and the candidate chaotic activities are b, c and x. Finally, the chaotic activity is x.

The program code is as follows:

Algorithm 1: Direct Chaotic Activity Filtering

Input: event log L
Output: event log list Q
$L'=L$
$Q=\langle L'\rangle$
While $|Activities(L')|>2$
 $acts=Activities(L')$
 $counts\leftarrow\emptyset, countms\leftarrow\emptyset, countns\leftarrow\emptyset, countmns\leftarrow\emptyset$
 for i **in** $Activities(L')$ **do**
  $count1\leftarrow0, countm\leftarrow0, countn\leftarrow0, countmn\leftarrow0$
  for j **in** $Activities(L')$ **do**
   if($D_{ij}.d_1>0$) **then**
    $count1$++
   if($D_{ij}.d_2>0$) **then**
    $count1$++
   if($D_{ij}.d_1>0$ and $D_{ij}.d_2>0$) **then**
    countm++
    if($|D_{ij}.d_1-D_{ij}.d_2|>0$) **then**
     $countn$++
     if($countm$!=0) **then**
      $countmn=countn/countm$
  end for
  $counts\leftarrow\{counts\cup(i,count1)\}$
  $countms\leftarrow\{countms\cup(i,countm)\}$
  $countns\leftarrow\{countns\cup(i,countn)\}$
  $countmns\leftarrow\{countmns\cup(i,countmn)\}$
  end for
  $rem\leftarrow\emptyset$
  for a **in** $Activities(L')$ **do**
   if($counts$.get(a)>$average$($counts$) and $countms$.get(a)>$average$($countms$)) **then**
    if($countns$.get(a)>$average$($countns$) and $countmns$.get(a)>$average$($countmns$)) **then**
     $rem\leftarrow\{rem\cup a\}$
  end for
  $L'=L'\uparrow acts\backslash rem$
  $Q=Q\cdot\langle L'\rangle$
end while
return Q

3.3. Indirect Chaotic Activity Filtering

Another way of the method proposed in algorithm 1 is to filter out some activities in the event log to make other activities in the log less chaotic. The total chaos degree of the event log is defined as the sum of the chaos degrees of all activities in the event log, that is, $CH_i(L)=\sum_{y\in Activities(L)}CH_i(y)$. Algorithm 2 describes an algorithm for iteratively filtering the activities from the event log that greatly reduce the chaos of the log. Contrary to algorithm 1, algorithm 2 selects the activities to be filtered according to the total chaos of the log after deleting the activities.

Algorithm 2 obtains the total chaos degree value of the filtered event log by calculating the filtered relation table between activities, and identifies chaotic activities in the original event log according to this value. Firstly, the activities in the log are deleted from the input original event log. After each activity is deleted, the relation between activities in the log is re-extracted to form a new relation table. Then, the total chaos degree of the event log after deleting activities is calculated according to the new relation table. The total chaos degree of the log is influenced by various activities in the log, and chaotic activities have a great influence on the total chaos degree of the event log, and the purpose of chaotic activity filtering is mainly to reduce the chaos degree of the event log to improve the quality of the process model. Therefore, by calculating the change degree of the total chaos degree of the log after deleting certain activities, the influence degree

of the activity on the total chaos degree of the event log can be obtained, so as to obtain the corresponding activity set that meets the conditions, and then judge whether the activity is chaotic or not through the obtained activity set. In algorithm 1, four conditions are used to calculate the chaos degree of activities, and in algorithm 2, condition (4) in algorithm 1 is deleted from the calculation of the chaos degree of activities. The total chaos degree of the event log L' for the i-th condition after deleting activity $y \in Activities(L)$ in log L is defined as $CH_i^L(y)$, $CH_i^L(y) = CH_i(L') = CH_i(L \uparrow Activities(L) \setminus \{y\})$

Take log L=[<*a*,*b*,*c*,*x*>^{10},<*a*,*b*,*x*,*c*>^{10},<*a*,*x*,*b*,*c*>^{10}] as an example. For condition (1) ,take activity x as an example. After deleting activity *x*, the event log *L*' is *L*'=[<*a*,*b*,*c*>^{30}] . For the calculation of the relation between activities, #(<*a*,*b*>,*L*')=30, #(<*b*,*c*>,*L*')=30 are obtained, thus obtains the corresponding relation tables, for relation table to calculate, { $CH_1(a)$=1, $CH_1(b)$=2, $CH_1(c)$=1 } is obtained, so we get $CH_1^L(a)$=4. This calculation is performed for all activities in the original log, and the corresponding results of each activity are obtained, and the average value of all results is obtained. In this original log, the average value is 7.5. The calculated result of activity x is less than the average value, so it is judged that the candidate chaotic activity is x. For condition (2) , taking activity b as an example, after deleting activity b, the event log *L*' is *L*'=[<*a*,*c*,*x*>^{10},<*a*,*x*,*c*>^{20}], and the relation between activities is calculated to get #(<*a*,*c*>,*L*')=10, #(<*a*,*x*>,*L*')=20, #(<*c*,*x*>,*L*')=10, #(<*x*,*c*>,*L*')=20, thus obtains the corresponding relation tables, for relation table to calculate, {$CH_2(a)$=0, $CH_2(c)$=1, $CH_2(x)$=1} is obtained, So we get $CH_2^L(a)$=2. This calculation is performed for all the activities in the original log, and the corresponding results of each activity are obtained. The average value of all the results is obtained. In this original log, the average value is 2. The calculated result of activity x is less than the average value, so it is judged that the candidate chaotic activity is *x*. For condition (3), taking activity a as an example, after deleting activity a, the event log *L*' is *L*'=[<*b*,*c*,*x*>^{10}, <*b*,*x*,*c*>^{10}, <*x*,*b*,*c*>^{10}], and the relation between activities is calculated to get #(<*b*,c>,*L*')=20, #(<*b*,*x*>,*L*')=10, #(<*c*,*x*>,*L*')=10, #(<*x*,*b*>,*L*')=10, #(<*x*,*c*>,*L*')=10,thus, the corresponding relation table is obtained, For the relational table, the calculation for condition (3) gives { $CH_3(b)$=1, $CH_3(c)$=1, CH3(x)=2 } is obtained. So we get $CH_3^L(a)$=4. This calculation is performed for all the activities in the original log, and the corresponding results of each activity are obtained. The average value of all the results is obtained. In this original log, the average value is 2. The calculated result of activity *x* is less than the average value, so it is judged that the candidate chaotic activity is *x*.

Take the event log *L*=[<*a*,*b*,*c*,*x*>^{10},<*a*,*b*,*x*,*c*>^{10},<*a*,*x*,*b*,*c*>^{10}] as an example. For the condition (1), we obtain { $CH_1^L(a)$=10, $CH_1^L(b)$=8, $CH_1^L(c)$=8, $CH_1^L(x)$=4}, with an average of 7.5, so the candidate chaotic activity is x. For the condition (2), it is obtained that { $CH_2^L(a)$=4, $CH_2^L(b)$=2, $CH_2^L(c)$ =2, $CH_2^L(x)$=0}, the average value is 2, so the candidate chaotic activity is x. For condition (3), we obtain that { $CH_3^L(a)$=4, $CH_3^L(b)$=2, $CH_3^L(c)$=2, $CH_3^L(x)$=0}, the average value is 2, so the candidate chaotic activity is *x*. Finally, the chaotic activity is *x*.

The program code is as follows:

Algorithm 2: Indirect Chaotic Activity Filtering

Input: event log *L*
Output: event log list *Q*
L'=*L*
Q= 〈*L'*〉
While |Activities(L')|>2
 acts=Activities(L')
 countos←∅,countmos←∅,countnos←∅
 for *a* **in** Activity(*L'*) **do**
  L'=*L'*↑*acts*\{*a*}
  counts←∅,*countms*←∅,*countns*←∅
  for *i* **in** *Activities*(*L'*) **do**
   count1←0,*countm*←0,*countn*←0,*countmn*←0
   for *j* **in** *Activities*(*L'*) **do**

```
        if(D_ij.d_1>0) then
          count1++
        if(D_ij.d_2>0) then
          count1++
        if(D_ij.d_1>0 and D_ij.d_2>0) then
          countm++
          if(|D_ij.d_1-D_ij.d_2|>0) then
            countn++
      end for
      counts←{counts∪(i,count1)}
      countms←{countms∪(i,countm)}
      countns←{countns∪(i,countn)}
    end for
    counto←0,countmo←0,countno←0
    for ac in Activities(L')
      counto=counto+counto+counts.get(ac)
      countno=countno+countno+countns.get(ac)
    end for
      countos←{countos∪(i,counto)}
      countmos←{countmos∪(i,countmo)}
      countnos←{countnos∪(i,countno)}
  end for
  rem←∅
  for a in Activities(L') do
    if(countos.get(a)>average(countos) and
countmos.get(a)>average(countmos)) then
      if(countnos.get(a)>average(countnos)) then
        rem←{rem∪a}
  end for
  L'=L'↑acts\rem
  Q=Q·〈L'〉
 end while
return Q
```

4. Experimental Analysis

In this section, simulation event logs and real event logs were used to conduct comparative experiments on different chaotic activity filtering approaches, and the proposed chaotic activity filtering approaches were evaluated experimentally. Table II shows some main statistical data of real event logs. Tables III ~ V show the results of experiments using simulation logs, and Figs. 5~ 7 show the results of experiments using real event logs. The experimental results prove the effectiveness of the filtering approaches proposed in this paper, which can improve the running speed while ensuring the accuracy. The experiments were all based on PC Intel Core i7-5500U 2.40GHz CPU, 4 GB RAM environment and implemented in Java language.

4.1. Data Set and Experimental Setup

4.1.1. Simulation Log

Firstly, we can compare the event logs before and after filtering chaotic activities by simulating the event logs containing chaotic activities, and judge whether the filtering approach can identify all chaotic activities in the logs and calculate the number of normal activities deleted from the logs if all inserted chaotic activities are filtered out.

In order to verify the accuracy of the chaotic activity filtering approach, the event log containing chaotic activities is simulated. First, in step (1), a synthetic event log is generated from the process model, so that it can be determined that all activities of the model are not chaotic activities. Then, in step (2), artificially insert activities positioned at random positions in the log. Since the locations in these activity logs are randomly selected, it is assumed that these activities are chaotic. Change the number (k) of randomly-positioned activities inserted to assess how well the chaotic activity filtering techniques are able to deal with different numbers of randomly-positioned activities in the event log. In addition, the frequency of inserted randomly-positioned activities is changed, among which three types of random location activities are distinguished, and different randomly-positioned activity insertion methods are used to insert chaotic activities with different frequencies in the log:①$f(k)$: frequent randomly-positioned activities are inserted into the log, and the occurrence frequency of each of the k inserted randomly-positioned activities is set to $\max_{a\in Activities(L)}\#(a,L)$.②$i(k)$: insert infrequent randomly-positioned activities into the log, and set the frequency of each of the k inserted randomly-positioned activities as $\min_{a\in Activities(L)}\#(a,L)$. ③ $m(k)$: evenly randomly-positioned activities are insert into that log, and for each of the k randomly-positioned

activities, the occurrence frequency of the activities is randomly select from the uniform probability distribution U[$\min_{a \in Activities(L)} \#(a,L)$, $\max_{a \in Activities(L)} \#(a,L)$]. In step (3), the activities are removed by using different chaotic activity filtering approaches, and the randomly-positioned activities are filtered from the event log until all k artificially inserted activities are removed. Then, count the number of activities that were originally in the process model deleted from the log, calculate the error deletion rate(mdr) of the filtering approach, i.e., the number of deletion errors divided by the number of original activities, and calculate the time required to delete k artificially inserted activities (step (4)).

4.1.2. Real Log

Secondly, the real data was evaluated, in which case there is no basic factual knowledge about which activities in the process are chaotic. This motivates a more indirect evaluation, that is, after filtering out activities by using the proposed activity filtering technology, we evaluated the quality of the process model found in the event log and the time required to filter out chaotic activities in the log. Three real event logs are used to evaluate the proposed chaotic activity filtering approaches. **Table II** shows some main statistical data of these event logs.

Environmental permit: The dataset consists of five event logs that record the execution of the building permit application process in five different anonymous cities;

Sepsis: This event log contains sepsis case events from the hospital, with each track representing the course of treatment of one sepsis patient, about 1,000 cases, and a total of 15,000 events recorded across 16 different activities;

BPI Challenge 2012: The event log relates to the loan application process of financial institutions in the Netherlands. The cases in the log contain information about the main application and the various stages of the opposition process.

Real event logs used in the experiment.

Data set	Track number	Number of events	Activity number
Environmental permit	1434	8577	27
Sepsis	1050	15214	16
BPI Challenge 2012	13087	164506	23

4.2. Evaluation Indicators

4.2.1. Simulation Log

For the synthetic event log, the number of original activities that were mistakenly removed from the event log during the process of filtering out all manually inserted random activities is calculated, and the quality of chaotic activity filtering approach is evaluated by this method. If this filtering approach is used to filter out all the inserted chaotic activity errors and delete a small number of original activities, it indicates that the method has a good effect. The accuracy of the method is calculated by calculating the error deletion rate. At the same time, the running time of the filtering approach is calculated in the process of filtering chaotic activities.

4.2.2. Real Log

The purpose of chaotic activity filtering is to improve the quality of process model, so the effectiveness of chaotic activity filtering approach can be evaluated by evaluating the quality of process model. Firstly, the chaotic activity filtering technology is used to filter the original event log to get the filtered log. Then, the filtered log is used for process discovery technology to get the process model. After that, the quality of the obtained process model is evaluated. If the quality of the process model is high, it shows that the chaotic activity filtering approach is effective.

In the evaluation of real data, because there is no information about chaotic activities in the log, it is necessary to use more indirect evaluation.

After filtering the chaotic activities in the event log by using the chaotic activity filtering approach mentioned above, we evaluate the quality of the process model found in the filtered event log. In this section, we evaluated the chaotic activity filtering technology by evaluating the quality of the discovered process model and calculated the running time of the filtering approach when active filtering was performed on each log.

There are several quantitative methods to evaluate the quality of the process model of event logs. Ideally, the process model M should contain all the behaviors that can be observed in the event log L, and this standard can be expressed as fitness. In addition, model M should not contain too many additional behavior that was not seen in the event log. This kind of standard is called precision. IM algorithm is used to mine the event log to get the model. For each process model we discovered, we measured the *fitness* and *precision* of the filtered log. We use alignment-based fitness measurement [12] to measure *fitness*, and we use negative event accuracy [13] to measure *precision*. Based on the results of *fitness* and *precision*, we also calculated *F-score* [15], that is, the harmonic average between *fitness* and *precision*.

$$F - score = 2 \cdot \frac{precison \cdot fitness}{(precison + fitness)} \quad (2)$$

The direct filtering approach and indirect entropy-based filtering approach proposed in this paper are compared with the traditional direct entropy-based filtering approaches and indirect entropy-based filtering approaches respectively.

4.3. Experimental Results

4.3.1. *Simulation Log*

The frequency and time of chaotic activities in the event log are uncertain, so we can simulate chaotic activities in the log by inserting randomly-positioned activities into the log. Randomly-positioned activities are inserted randomly in the log, and randomly-positioned activities with different frequencies can be obtained through different insertion methods.

First of all, experiments were carried out on the synthetic event log with k randomly-positioned activities inserted, and the results are as follows.

Table III shows the experimental results of inserting frequent randomly-positioned activities into the log. k frequent randomly-positioned activities were inserted into the log by using the frequent randomly-positioned activity insertion method $f(k)$, and the frequencies of the k randomly-positioned activities were all the frequencies of the activities with the highest frequency in the log.

Table IV shows the experimental results of inserting uniform randomly-positioned activities into the log. k uniform randomly-positioned activities were inserted into the log by using the uniform randomly-positioned activity insertion method $m(k)$, and the frequency of each of the k inserted randomly-positioned activities was randomly selected from the uniform probability distribution with minimum value min $_{a \in Activities(L)}\#(a,L)$ and maximum value max $_{a \in Activities(L)}\#(a,L)$.

Table V shows the experimental results of inserting infrequent randomly-positioned activities into the log. k infrequent randomly-positioned activities were inserted into the log by using the infrequent randomly-positioned activity insertion method $i(k)$, and the frequencies of k randomly-positioned activities were the frequencies of the activity with the lowest frequency in the log. The experimental results show the number of original activities wrongly deleted in the process of deleting all artificially inserted randomly-positioned activities. This method is used to evaluate the effect of the filtering approach. If the number of original activities lost is large, it indicates that the accuracy of the method for identifying chaotic activities is low.

In the experiment, the Direct(dfr) and Indirect(dfr) approach proposed in this paper were compared with the four methods proposed in reference [14] (direct entropy-based activity filtering approach, direct entropy-based activity filtering approach with Laplace smoothing, indirect entropy-based activity filtering approach, and indirect entropy-based activity filtering approach with Laplace smoothing) to compare the running time and error deletion rate of the approaches, that is, the accuracy of the filtering approaches. Through the experimental results, we can see that

although the results obtained for different frequencies and different amounts of random activities are different, the filtering approaches proposed in this paper can reduce the running time without losing the accuracy.

TABLE II. The error deletion rate(*mdr*) and runtime per filtering approach(Frequent) (ms).

Approach	*f*(1)		*f*(2)		*f*(3)		*f*(4)		*f*(5)		*f*(6)		*f*(7)	
	mdr	*time*	*mdr*	*time*	*mdr*	*time*	*mdr*	*time*	*mdr*	*time*	*mdr*	*time*	*mdr*	*time*
Direct	0	44	0	48	0	56	0	60	0	68	0	73	0	88
Direct(α=1/|A|)	0	53	0	57	0	58	0	63	0	70	0	79	0	96
Indirect	0	89	0	105	0	114	0	126	0	138	0	156	0	168
Indirect(α=1/|A|)	0	96	0	111	0	120	0	136	0	153	0	172	0	184
Direct(dfr)	*0*	*35*	*0*	*39*	*0*	*45*	*0*	*50*	*0*	*57*	*0*	*65*	*0*	*71*
Indirect(dfr)	*0*	*62*	*0*	*78*	*0*	*81*	*0*	*104*	*0*	*112*	*0*	*113*	*0*	*136*

TABLE III. The error deletion rate(*mdr*) and runtime per filtering approach(Uniform) (ms).

Approach	*m*(1)		*m*(2)		*m*(3)		*m*(4)		*m*(5)		*m*(6)		*m*(7)	
	mdr	*time*	*mdr*	*time*	*mdr*	*time*	*mdr*	*time*	*mdr*	*time*	*mdr*	*time*	*mdr*	*time*
Direct	0	42	0	43	0	48	0	49	0	58	0	69	0	81
Direct(α=1/|A|)	0	51	0	54	0	56	0	60	0	66	0	75	0	82
Indirect	0.143	82	0	95	0	107	0	129	0	136	0	138	0	158
Indirect(α=1/|A|)	0	86	0	103	0	113	0	129	0	142	0	165	0	178
Direct(dfr)	*0*	*34*	*0*	*37*	*0*	*40*	*0*	*46*	*0*	*52*	*0*	*58*	*0*	*63*
Indirect(dfr)	*0*	*58*	*0*	*71*	*0*	*78*	*0*	*92*	*0*	*97*	*0*	*109*	*0*	*121*

TABLE IV The error deletion rate(*mdr*) and runtime per filtering approach(Frequent) (ms).

Approach	*i*(1)		*i*(2)		*i*(3)		*i*(4)		*i*(5)		*i*(6)		*i*(7)	
	mdr	*time*	*mdr*	*time*	*mdr*	*time*	*mdr*	*time*	*mdr*	*time*	*mdr*	*time*	*mdr*	*time*
Direct	0	38	0	40	0	42	0	44	0	54	0	59	0	67
Direct(α=1/|A|)	0	46	0	49	0	51	0	56	0	60	0	67	0	72
Indirect	0.143	79	0	90	0.143	104	0.143	111	0.143	121	0.143	126	0.143	141
Indirect(α=1/|A|)	0	82	0	93	0	107	0	116	0	128	0	135	0	146
Direct(dfr)	*0*	*31*	*0*	*34*	*0*	*37*	*0*	*39*	*0*	*48*	*0*	*54*	*0*	*57*
Indirect(dfr)	*0*	*55*	*0*	*63*	*0*	*76*	*0*	*87*	*0*	*94*	*0.143*	*98*	*0*	*116*

4.3.2 Real Log

Experiments were carried out with real event logs to compare the accuracy and running time of the method proposed in this paper with the four methods proposed in reference [13] and the Least−frequent−first chaotic activity filtering approach. In experiments with real event logs, the quality of chaotic activity filtering technology was evaluated by an indirect method, that is, evaluated the quality of the model obtained after process mining of the event logs obtained after chaotic activity filtering had been completed The higher the quality of the model, the more effective the method is. The model evaluation method used in this section is to calculate the *F-score* value of the model.

Taking the event log Sepsis as an example, **Fig. 2** is the directly-follows graph of the real event log Sepsis used in the experiment, and **Figs. 3** and **4** are the directly-follows graphs of the event log obtained by filtering the chaotic activity of the real event log Sepsis. According to **Fig. 3**, the event log Sepsis is filtered by the direct chaotic activity filtering approaches proposed in this paper to get three filtered logs, and the number of remaining activities in the logs is 9 and 7 in turn; According to **Fig. 4**, the event log Sepsis is filtered by the indirect chaotic activity filtering approaches proposed in this paper to get three filtered logs, and the number of remaining activities in the logs is 8 and 7 in turn.

Multiple filtered event logs can be obtained from original event logs after chaotic activity filtering. **Fig. 2** to **Fig. 4** show the directly-follows graphs of all filtered event logs obtained after the real event log are filtered. As can be seen from the directly-follows graphs, the chaos degree of the directly-follows relation between activities in the log is obviously improved after being filtered by chaotic activities. And as the number of activities decreases, the disorder of the directly-follows relation between activities decreases. In addition, according to the conditions proposed in this paper for the identification of chaotic activities, the proposed chaotic activity filtering approach can stop filtering when the number of activities deleted from the log reaches a certain degree, which is related to the log.

As can be seen from the figures, the chaotic activity filtering algorithm proposed in this paper can filter multiple chaotic activities from the event log in each run, so the algorithm can improve the efficiency of the chaotic activity filtering approach. And the number of remaining activities in the log can be controlled within a reasonable range, so the algorithm can ensure the integrity of the event log to a certain extent. After filtering the chaotic activities in the event log, it can be seen by observing the directly-follows graphs of the filtered logs that the degree of chaos between the activities in the log is effectively reduced. Therefore, these approaches can effectively reduce the degree of chaos in the event log and improve the quality of the model.

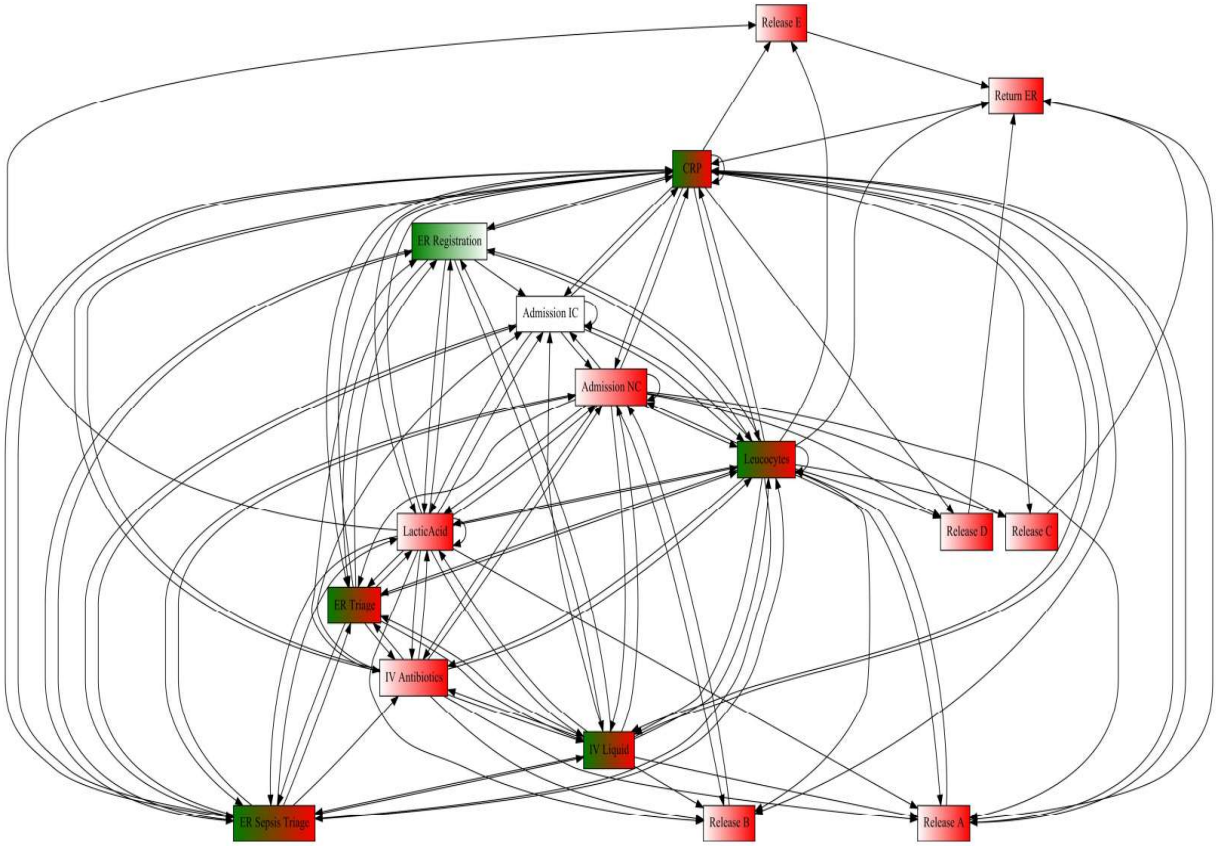


Fig. 2. Directly-follows graph for Sepsis

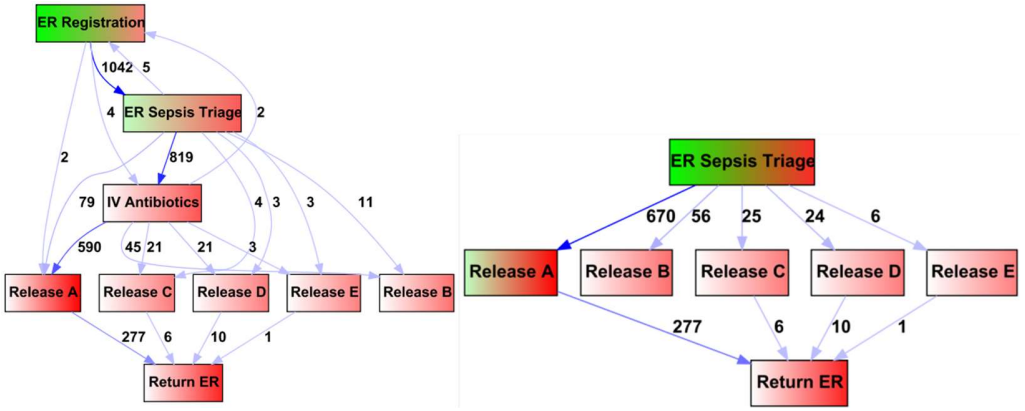


Fig. 3. Directly-follows graph for Sepsis-direct

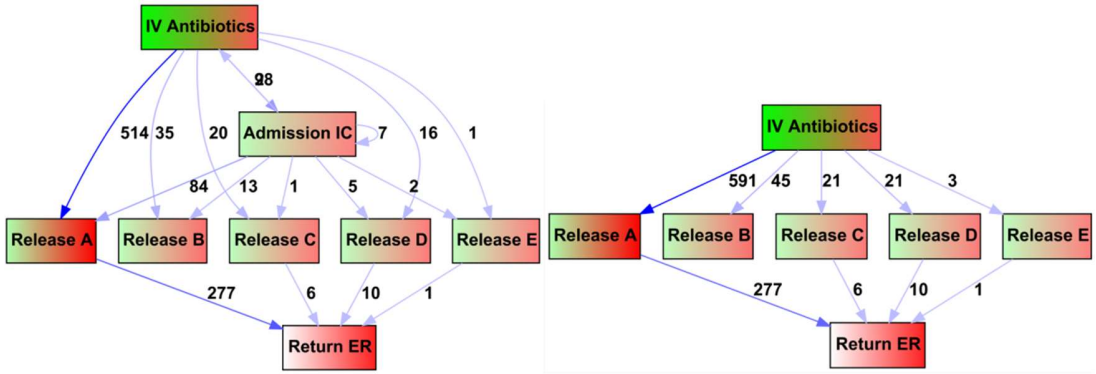


Fig. 4. Directly-follows graph for Sepsis-indirect

The experimental results are shown in **Fig. 5~ Fig. 7**, and each figure consists of two parts. Part (a) shows the comparison of model quality changes with the reduction of the number of remaining activities in the event log after filtering, that is, the accuracy comparison of filtering approaches. Part (b) shows the comparison of the time required for filtering chaotic activities in the event log. **Fig. 5** shows the experimental results obtained by comparing the effectiveness and efficiency of the traditional chaotic activity filtering approaches and the chaotic activity filtering approaches proposed in this paper using BPI Challenge 2012, **Fig. 6** shows the results of a comparison experiment with the traditional chaotic activity filtering approaches and the chaotic activity filtering approaches proposed in this paper in terms of effectiveness and efficiency using Sepsis. **Fig. 7** shows the results of the comparison experiment between the traditional chaotic activity filtering approaches and the chaotic activity filtering approaches proposed in this paper in terms of effectiveness and efficiency

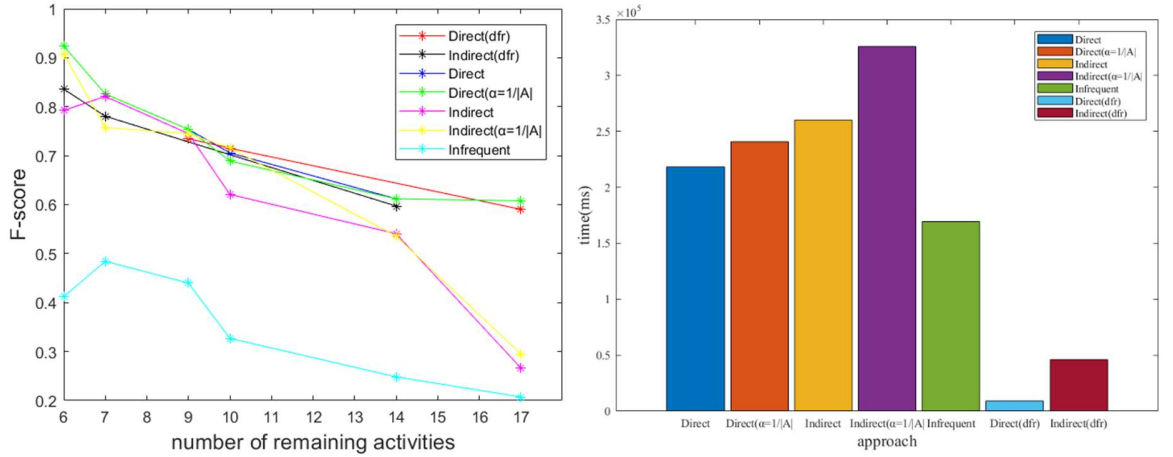
Different chaotic activity filtering approaches are used to filter the event log, and then the filtered event log is used to obtain the process

model through the process discovery algorithm. The quality of the process model is used to evaluate the accuracy of the chaotic activity filtering approach. Thus, the chaotic activity filtering approaches proposed in this paper has high accuracy. Moreover, through observation, it can be found that with the decrease of the number of activities in the event log, the quality of the process model is gradually improved. For different event logs, the performance of chaotic activity filtering approaches is different.

By observing the accuracy comparison results of the following chaotic activity filtering approaches, the following conclusions can be obtained. Firstly, from the experimental results in **Fig. 5 (a)** to **Fig. 7(a)**, it can be concluded that the chaotic activity filtering approaches proposed in this paper is superior to the frequency-based chaotic activity filtering approach in accuracy, and can maintain the integrity of the filtered event log to a certain extent. The traditional chaotic activity filtering approaches deletes the identified chaotic activities from the log successively until there are two remaining activities in the log, so it is difficult to ensure the integrity of the filtered log. However, the approaches proposed in this paper identifies and filters chaotic activities in logs by using the method of calculating thresholds, which can ensure that there are still more activities in event logs after filtering, so that the logs can maintain certain integrity after filtering, and thus maintain the integrity of the process model. From the experimental results in **Figs. 5(b)** to **7(b)**, it can be seen that in terms of chaotic activity filtering for real event logs, the proposed chaotic activity filtering approaches takes less time than the corresponding traditional chaotic activity filtering approaches. Therefore, the filtering approaches proposed in this paper can obtain a suitable process model at a faster speed.

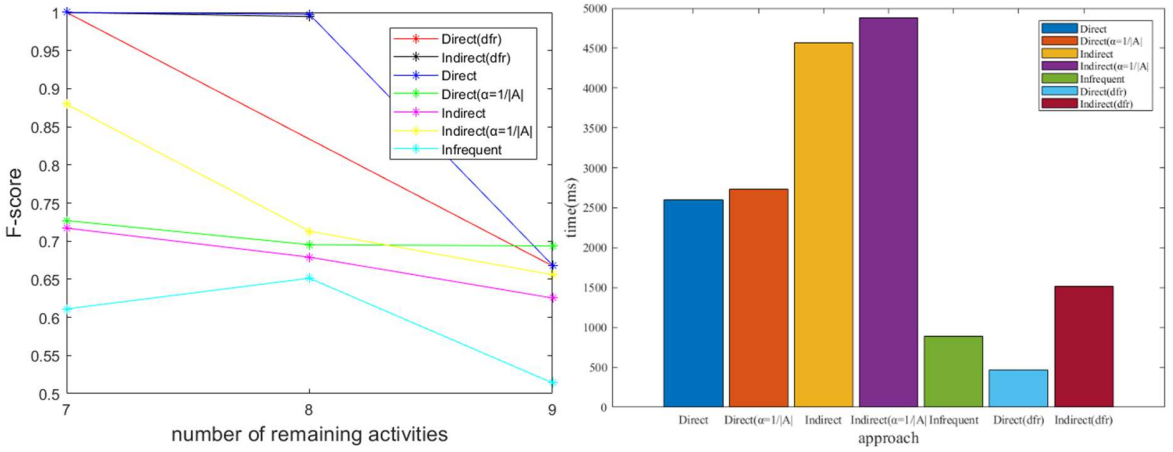
As for the time performance of chaotic activity filtering approaches, the time required to filter real event logs by different chaotic activity filtering approaches is calculated, and the time performance of chaotic activity filtering approaches proposed in this paper is compared with four approaches proposed in reference [13] and least−frequent−first chaotic activity filtering approaches. From the experimental results in **Fig. 5 (b)** to **Fig. 7(b)**, it can be seen that the time of the chaotic activity filtering approaches proposed in this paper is less than the corresponding traditional chaotic activity filtering approaches for real event logs. Therefore, the chaotic activity filtering approaches proposed in this paper has certain high efficiency, which can effectively shorten the time required for chaotic activity filtering and improve the efficiency of log preprocessing.

It can be seen from the experimental results that the effect of the filtering approach is closely related to the event log itself, so different results can be obtained for different event logs. However, comparing the approaches proposed in this paper with other approaches, the approaches proposed in this paper can reduce the running time without losing too much accuracy and ensure certain integrity of the log.



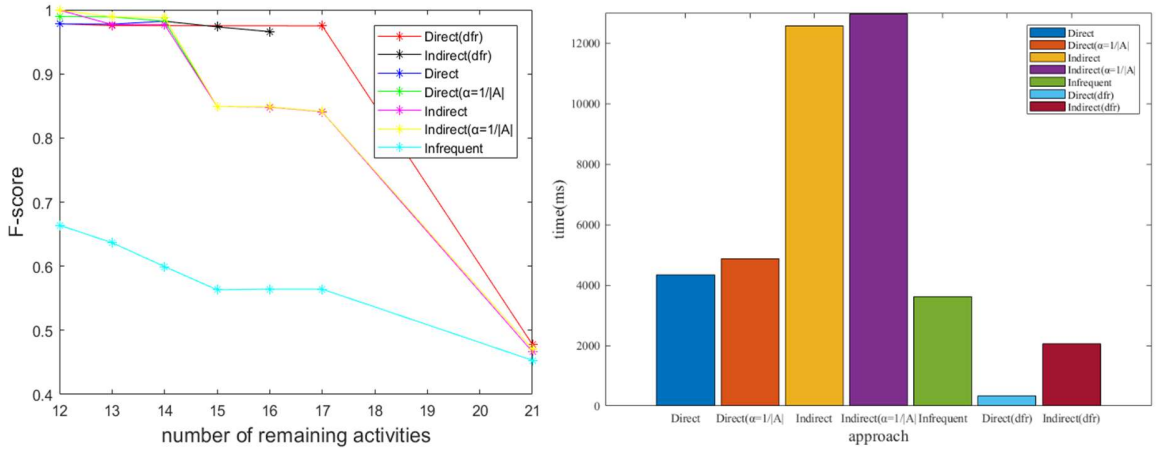
(a) model quality comparison (b) comparison of running time

Fig. 5. Experimental result of BPI Challenge 2012



(a) model quality comparison (b) comparison of running time

Fig. 6. Experimental result of Sepsis



(a) model quality comparison (b) comparison of running time

Fig. 7. Experimental result of Environmental permit

5. Conclusion

In this paper, approaches of filtering chaotic activities by using the directly-follows relation between activities has been proposed. The approaches can identify and filter chaotic activities by extracting the relations between activities in the event log. The approaches can identify and delete multiple activities in each algorithm run by using the judgment conditions. Compared with other approaches, it can reduce the time of preprocessing the event log and improve the efficiency without losing too much accuracy. And in general, the approaches can ensure the integrity of the filtered event log. At the same time, the filtering approaches proposed in this paper can also reduce the number of logs obtained after filtering the original logs, so it can shorten the time required for indirect evaluation of chaotic activity filtering approaches. By using simulation log and real event log to test, it can be concluded that the method proposed in this paper has certain effectiveness and high efficiency. However, the algorithm proposed in this paper reduces the running time, and at the same time, the accuracy of chaotic activity identification is partially lost, and the running effect of the algorithm is related to the event log itself. In the future, we can consider trying to set up new chaotic activity judgment conditions, such as association rules between activities, to achieve a balance between accuracy and running time.

6. Acknowledgement

The authors would like to thank anonymous referees and the editor for their carefully reading the paper and f for their insightful comments and suggestions. This research is supported by the Shandong Provincial Undergraduate Teaching Reform Project (Grant Number: Z2021450) & the Shandong Provincial Natural Science Foundation of P.R China (Grant Number: ZR2020QF069) & Graduate Education and Teaching Reform Research Project, Shandong University of Technology (Grant Number: 1172544051).

References

[1] Conforti R,Rosa M L,Hofstede A H M T.Filtering Out Infrequent Behaviour from Business Process Event Logs[J].IEEE Transactions on Knowledge & Data Engineering,2017,29(2):300-314.

[2] Lu X,Fahland D,van den Biggelaar F J H M,van der Aalst W.M.P.Detecting deviating behaviors without models[C].Proceedings of the international workshop on business process intelligence.2015:126–139.

[3] Sani M F, van Zelst S J, van der Aalst W M P. Improving process discovery results by filtering outliers using conditional behavioural probabilities[C] In Proceedings of the international workshop on business process intelligence 2018: 216-229 ..

[4] Van der Aalst W, Weijters T, Maruster L. Workflow mining: Discovering process models from event logs[J]. IEEE transactions on knowledge and data engineering, 2004, 16(9): 1128-1142.

[5] Leemans S J J,Fahland D, van der Aalst W M P. Discovering block-structured process models from event logs - a constructive approach[C] . In International conference on applications and theory of petri nets and concurrency,2013:311-329.

[6] Leemans S J J,Fahland D, van der Aalst W M P.Discovering Block-Structured Process Models from Event Logs Containing Infrequent Behaviour[C] International Conference on Business Process Management,2013:66-78.

[7] Ghionnal L,Greco G,Guzzo A,et al.Outliner Detection Techniques for Process Mining Application[C] Foundations of Intelligent Systems, International Symposium,Ismi2008,Toronto,Canada,May20-23,2008,Proceedings.DBLP,2008:150-159.

[8] Cheng H J, Kumar A. Process mining on noisy logs—Can log sanitization help to improve performance? [J]. Decision Support Systems, 2015, 79: 138-149.

[9] Van Dongen B F, de Medeiros A K A, Verbeek H M W, et al. The ProM framework: A new era in process mining tool support[C].In International conference on application and theory of petri nets, 2005: 444-454.

[10] Leemans S J J, Fahland D, Van Der Aalst W M P. Process and Deviation Exploration with Inductive Visual Miner[J]. BPM (demos), 2014, 1295(8).

[11] Adriansyah A, van Dongen B F, van der Aalst W M P. Conformance checking using cost-based fitness analysis[C] .In Proceedings of the 15 IEEE international enterprise distributed object computing conference (EDOC). IEEE, 2011: 55-64.

[12] vanden Broucke S K L M, De Weerdt J, Vanthienen J, et al. Determining process model precision and generalization with weighted artificial negative events[J]. IEEE Transactions on Knowledge and Data Engineering, 2013, 26(8): 1877-1889.

[13] De Weerdt J, De Backer M, Vanthienen J, et al. A robust F-measure for evaluating discovered process models[C]. In Proceedings of the IEEE symposium on computational intelligence and data mining (CIDM). IEEE, 2011: 148-155.

[14] Măruşter L, Weijters A J M M, Van Der Aalst W M P, et al. A rule-based approach for process discovery: Dealing with noise and imbalance in process logs[J]. Data mining and knowledge discovery, 2006, 13: 67-87.

[15] De Weerdt J, De Backer M, Vanthienen J, et al. A robust F-measure for evaluating discovered process models[C]. In Proceedings of the IEEE symposium on computational intelligence and data mining (CIDM). IEEE, 2011: 148-155.