

Article

Pre-Processing Event Logs by Chaotic Filtering Approaches Based on the Direct Following Relationship

Tengzi Lv, Xiugang Gong *, Na Gong and Kaiyu Li

School of Computer Science and Technology, Shandong University of Technology, Zibo 255049, China; lvtz686@gmail.com (T.L.); gn874682003@outlook.com (N.G.); 20110506044@stumail.sdut.edu.cn (K.L.)

* Correspondence: gong_xg@sdut.edu.cn

Abstract: Process discovery aims to discover process models from event logs to describe actual business processes. The quality of event logs has an impact on the quality of process models, so preprocessing methods can be used to improve the quality of event logs. 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 model 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 makes it difficult to balance accuracy and time performance. Therefore, a direct method for filtering chaotic activities is proposed in this paper. By analyzing the relationship between activities, chaotic activities are identified in the log according to the characteristics of chaotic activities and the direct following relationship of activities as the judgment condition, and the filtering of chaotic activities in the event log is realized. In addition, this paper proposes an indirect chaotic activity filtering method, which identifies and filters chaotic activities in the log by analyzing the influence of the existence of different activities on the overall chaos degree of the log. The proposed method is compared with the traditional chaotic activity filtering method 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 method. By summarizing the experimental results, it is found that the accuracy of the proposed chaotic activity filtering methods is greater than that of the frequency-based filtering method and is close to that of the entropy-based chaotic activity filtering methods. Moreover, compared with other filtering methods used in the experiment, the chaotic activity filtering method proposed in this paper can improve the efficiency by 23.4% on average for simulation logs, and by 84.25% on average for real event logs. It is concluded that compared with other filtering methods, the proposed chaotic activity filtering methods have higher accuracy and can effectively improve the time performance of chaotic activity filtering. Therefore, the chaotic activity filtering method proposed in this paper can balance the accuracy and time performance, and can ensure the integrity of the filtered event log to a certain extent.

Keywords: process mining; chaotic activities; pre-processing; process model; direct following relationship; event log



Citation: Lv, T.; Gong, X.; Gong, N.; Li, K. Pre-Processing Event Logs by Chaotic Filtering Approaches Based on the Direct Following Relationship. *Appl. Sci.* **2024**, *14*, 6994. <https://doi.org/10.3390/app14166994>

Academic Editor: Christos Bouras

Received: 3 July 2024

Revised: 26 July 2024

Accepted: 4 August 2024

Published: 9 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Process mining [1] integrates various business processes by modeling, managing, controlling, and optimizing the entire lifecycle of the business process. Research on process mining focuses on analyzing 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 a business process model through the information contained in event logs. However, in the actual process of process execution, there are some random

positioning activities that do not belong to the process itself, which are called chaotic activities. The process model is obtained by process discovery technology. The existence of chaotic activities greatly affects the quality of the process model. The occurrence probability of chaotic activities is not affected by other activities, and the occurrence of chaotic activities does not change the occurrence probability of other activities. Therefore, when there is chaotic activity in the event log, the process of the process model found from the event log is very complex, and the existence of chaotic activity will lead to an excessive generalization of the resulting model.

In practical scenarios, to obtain good quality process models, it is important to preprocess the event log, that is, to improve the quality of the model by identifying activities, events, or traces that affect the quality of the process model and deleting them. Similar event log preprocessing methods also include filtering noise and low-frequency activities. The traditional filtering method filters the chaotic activity in the event log by filtering the noise or infrequent activity, but these methods have some shortcomings. It has been proposed in the literature [2] that chaotic activities can be identified and filtered by calculating activity entropy. However, this method calculates the entropy of each activity by traversing each activity in the log, and then identifies and removes one chaotic activity in each filtering step through the entropy of each activity until there are two activities left in the log. Because this method uses an iterative method to delete the most chaotic activities from the log in each round of the filtering algorithm, and the calculation of entropy is complex, the process of chaotic activity filtering for event logs containing many trajectories and activities takes a lot of time, and the efficiency of this method is low in chaotic activity filtering.

In this paper, new chaotic activity filtering approaches based on the direct following relationship between activities are proposed, and chaotic activities are identified by using the algorithm to count the direct following relationship between each activity and other activities. Firstly, the direct following relationship between activities in the log is calculated, and then the 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 a direct following relationship, 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 the work of process mining, most of the existing methods of filtering event logs adopt the following methods: first, the behavior that occurs less frequently in the event log and has a certain impact on the model obtained through process mining is identified, and then the behavior is deleted to improve the quality of the process model. At present, there are four kinds of filtering methods to filter chaotic activities: event filtering technology, process discovery technology with a 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. Conforti et al. [3] proposed using an integer linear programming solver to construct prefix automata of event logs, and to remove infrequent arcs from the minimal prefix automata. Lu et al. [4] proposed a method for filtering abnormal events in event logs, which differentiates between events that are part of the main current behavior of the process and abnormal events by using event mapping and filters out abnormal events in the logs. Fani Sani et al. [5] proposed distinguishing events belonging to the main current behavior from abnormal events by using sequential pattern mining technology. van Zelst, S J et al. [6] proposed a universal event flow filter that relies on incremental update automata to filter out spurious events, aiming to detect and remove infrequent behavior from the event stream.

Process discovery technology with built-in filtering mechanism. Some process discovery techniques can delete infrequent elements when mining models to obtain high-quality models. In addition to a direct following relationship, Heuristic Miner [7] also defines the eventual following relationship between activities and filters out unusual direct following relationship and eventual following relationship. The eventual following relationship is different from the direct following relationship and is not affected by chaotic activities. Inductive Miner [8] is a process discovery algorithm, which first finds a direct following graph from the event log, and then discovers the process model in the second step. Inductive Miner in frequency (IMf) [9] 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. Fuzzy Miner [10] discovers process models by extracting final follow relationships from event logs that are not affected by chaotic activity. Benevento, E. et al. [11] used an innovative process mining (PM) approach called Interactive Process Discovery (IPD), which combines domain knowledge with available data.

Trace filtering technology. The purpose of trace filtering technology is to identify and delete the traces that affect the quality of the model. Ghionna et al. [12] proposed mining frequent patterns from logs and applying an MCL clustering algorithm to traces. Traces that are not allocated to clusters by the MCL clustering algorithm are regarded as abnormal traces and filtered from event logs. Cheng and Kumar [13] proposed a measure of supervision to filter out abnormal traces from event logs, assuming that there is a sub-log that has been manually checked and marked. This process involves checking and marking clean and noisy traces in sub logs, extracting classification rules using the PRISM rule induction algorithm, training labeled sub logs, and then applying these classification rules to identify and filter noisy traces from unlabeled sub logs. Koschmider, A. et al. [14] studied various aspects related to (semi-) automatic outlier/noise detection, and proposed a method to identify abnormal tracks in event logs based on clustering similarity between different tracks.

Activity filtering technology. The activity filtering method is to identify and delete infrequent activities from event logs to obtain a high-quality process model. Yi Guo et al. [15] proposed a method to filter chaotic activities in event logs by analyzing the bidirectional causal dependence between the model and the event log. Leemans et al. proposed Inductive Visual Miner [16], which is an interactive process discovery instrument that can use sliders to filter event logs and use the Inductive Miner algorithm for process discovery. Tax N proposed a new chaotic activity filtering technique that determines whether an activity is chaotic by calculating its entropy, and uses direct or indirect methods to filter chaotic activities in event logs. Lamghari, Z. et al. propose a technique that uses unsupervised learning to identify chaotic activities without labeling training data [17]. Lu et al. proposed a maximum probability path analysis algorithm based on the strong transfer relationship between activity distribution state and behavior [18]. Conditional probability entropy is used to pre-process infrequent logs to remove individual noisy activities that are very irregularly distributed in the tracks. The valid sequence is then extracted from the log based on the state transition information of the activity. Li et al. proposed an entropy-based approach filtering method for chaotic activity behavior [19], realized the identification of suspicious chaotic activity set by Laplace entropy, and built a query model based on logs containing suspicious chaotic activity.

Most existing chaotic activity filtering uses methods to remove noise [20] or to identify and filter chaotic activities based on their frequency of occurrence, such as the method proposed in references [3–5,9,11,12,14,18] and other filtering methods. Nonetheless, these approaches may lead to more deletion or missing deletion of activities, which cannot solve the problem of chaotic activities affecting the quality of event logs. However, the time complexity of chaotic activity filtering based on activity entropy is high, so the efficiency of chaotic activity filtering is low. Therefore, this study proposes chaotic activity filtering methods based on direct following relationships between activities. These methods filter out chaotic activities by statistically analyzing the relationships between each activity

and other activities in the event log. The proposed chaotic activity filtering methods are based on the direct following relationship between activities in the log, and do not involve complicated calculation. These methods determine which activities in the log are chaotic activities by setting thresholds, and filter multiple selected chaotic activities in each run. By using simulated event logs and real event logs, it can be seen that the chaotic activity filtering methods proposed in this paper can reduce the running time by at least 10%. Therefore, these methods can reduce the time required for chaotic activity filtering, effectively improve the efficiency of chaotic activity filtering, and effectively improve the quality of logs while ensuring their accuracy.

3. Chaotic Activity Filtering Approach Based on Direct Following Relationship

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 Direct Following Relationship

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 obtain the filtered event log. In the third stage, the process model is obtained from the filtered logs using the process discovery algorithm. This article proposes the chaotic activity filtering methods for business processes based on direct following relationships between activities. As shown in Figure 1, firstly, taking the event log with chaotic activities as input, extract the direct following relationship between activities in the event log and the activity set of the log. Then, based on the following relationship between each activity and other activities in the event log, determine which activities in the log are chaotic activities. Finally, remove the identified chaotic activities from the activity set of 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.

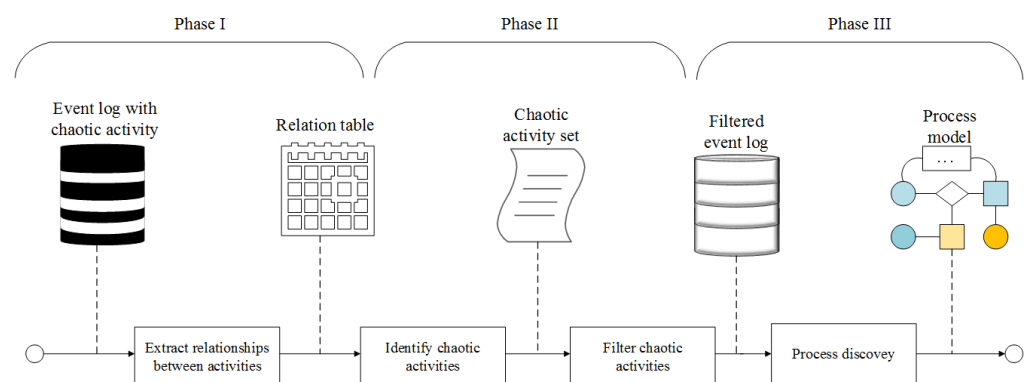


Figure 1. Chaotic activity filtering based on the direct following relationship.

Definition 1 (event, trace, event log). $L = [\langle a, b, c \rangle^2, \langle b, a, c \rangle^3]$ is an example event log of the process activity set $\{a, b, c\}$, which is composed of two traces $\langle a, b, c \rangle$ and three traces $\langle b, a, c \rangle$. $ActSet(L)$ represents the set of process activities that occur in L , for example, $ActSet(L) = \{a, b, c\}$. $\#(a, L)$ represents the number of occurrences of activity a in log L , for example, $\#(a, L) = 5$.

Extend the function $\#(a, L)$ to $\#(\sigma, L)$ to count the number of occurrences of a sequence in the log.

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

Definition 2 (multiple sets). $X = \{a_1, a_2, \dots, a_n\}$ denotes a finite set. $X \setminus Y$ represents a set of elements in set X but not in set Y , for example, $\{a, b, c\} \setminus \{a, c\} = \{b\}$. X^* represents the set of all finite sequences on set X . $\sigma = \langle a_1, a_2, \dots, a_n \rangle$ denotes a sequence of length n , where $\sigma(i) = a_i$ and $\langle \rangle$ is an empty sequence. $\sigma \upharpoonright X$ is the projection of σ on X , such as $\langle a, b, c, a, b, c \rangle \upharpoonright \{a, c\} = \langle a, c, a, c \rangle$. $\sigma_1 \cdot \sigma_2$ 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 (relationship table). Let D be a relationship 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 $d_{fs} = \#(i, j)$ be the direct following frequency between activities i and j , that is, the number of occurrences of j directly following i , and $d_{ps} = \#(j, i)$ be the direct preceding frequency between activities i and j , that is, the number of occurrences of i directly following j . Take the event log $L = [\langle a, b, c, d, x \rangle^{10}, \langle a, b, x, c, d \rangle^{10}, \langle a, x, b, c, d \rangle^{10}, \langle a, x, b, c, d \rangle^{10}]$ as an example. Table 1 shows the relationship between activities in the log.

Table 1. The relationship table of $L = [\langle a, b, c, d, x \rangle^{10}, \langle a, b, x, c, d \rangle^{10}, \langle a, x, b, c, d \rangle^{10}, \langle a, x, b, c, d \rangle^{10}]$.

Activity	<i>a</i>		<i>b</i>		<i>c</i>		<i>d</i>		<i>x</i>	
	d_{fs}	d_{ps}	d_{fs}	d_{ps}	d_{fs}	d_{ps}	d_{fs}	d_{ps}	d_{fs}	d_{ps}
<i>a</i>	0	0	30	0	0	0	0	0	10	0
<i>b</i>	0	30	0	0	30	0	0	0	10	10
<i>c</i>	0	0	0	30	0	0	20	0	10	10
<i>d</i>	0	0	0	0	0	30	0	0	10	10
<i>x</i>	0	10	10	10	10	10	10	10	0	0

3.2. Direct Chaotic Activity Filtering

Algorithm 1 describes an algorithm that uses an iterative approach to filter chaotic activity. The algorithm generates a list of event logs and takes the original event log as the starting element of the list. Each component in the queue includes a filtered version of L , and compared with the previous element, each subsequent component of the queue has supplementary 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 relationship table, and then the relation table is used to count 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 direct following relationship between chaotic activities and other activities is disordered. Therefore, in the event log, the direct following relationship and direct preceding relationship between chaotic activities and other activities are confusing, i.e., chaotic activities and most activities in the log have different following relations. Therefore, we can obtain 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 $y \in ActSet(L)$ in the log for the i -th condition as $CHi(y)$.

The program code is as follows:

Algorithm 1: Direct Chaotic Activity Filtering**Input:** event log L **Output:** event log list QLS

```

1:  $L' \leftarrow L$ 
2:  $QLS \leftarrow \langle L' \rangle$ 
3: While  $|ActSet(L')| > 2$ 
4:    $actset \leftarrow ActSet(L')$ 
5:    $counts \leftarrow \emptyset, countms \leftarrow \emptyset, countns \leftarrow \emptyset, countmns \leftarrow \emptyset$ 
6:   for  $i$  in  $ActSet(L')$  do
7:      $count1 \leftarrow 0, countm \leftarrow 0, countn \leftarrow 0, countmn \leftarrow 0$ 
8:     for  $j$  in  $ActSet(L')$  do
9:       if ( $D_{ij}.d_{fs} > 0$ ) then
10:         $count1++$ 
11:       if ( $D_{ij}.d_{ps} > 0$ ) then
12:         $count1++$ 
13:       if ( $D_{ij}.d_{fs} > 0$  and  $D_{ij}.d_{ps} > 0$ ) then
14:         $countm++$ 
15:       if ( $|D_{ij}.d_{fs} - D_{ij}.d_{ps}| > 0$ ) then
16:         $countn++$ 
17:       if ( $countm \neq 0$ ) then
18:         $Countmn \leftarrow countn / countm$ 
19:     end for
20:      $counts \leftarrow \{counts \cup (i, count1)\}$ 
21:      $countms \leftarrow \{countms \cup (i, countm)\}$ 
22:      $countns \leftarrow \{countns \cup (i, countn)\}$ 
23:      $countmns \leftarrow \{countmns \cup (i, countmn)\}$ 
24:   end for
25:    $rem \leftarrow \emptyset$ 
26:   for  $a$  in  $ActSet(L')$  do
27:     if ( $counts.get(a) > average(counts)$  and  $countms.get(a) > average(countms)$ ) then
28:       if ( $countns.get(a) > average(countns)$  and  $countmns.get(a) > average(countmns)$ ) then
29:          $rem \leftarrow \{rem \cup a\}$ 
30:   end for
31:    $L' \leftarrow L' \uparrow actset \setminus rem$ 
32:    $QLS \leftarrow QLS \cdot \langle L' \rangle$ 
33: end while
34: return  $QLS$ 

```

Condition (1): The degree of activity disorder is represented by the sum of the number of direct following relationships and the number of direct preceding relationships that exist between an activity and all other activities in the log, that is, for activity p , let the log be L , and the total of the number of elements in set S_1 and S_2 be counted as the degree of chaos in activity p . If the chaotic degree of activity p exceeds the set threshold, it is judged that p is a candidate chaotic activity.

$$S_1 = \left\{ q \mid \forall_{q \in ActiSet(L)} \#(< p, q >, L) > 0 \right\}, S_2 = \left\{ q \mid \forall_{q \in ActiSet(L)} \#(< q, p >, L) > 0 \right\}$$

Take the log $L = [<a,b,c,d,x>^{10}, <a,b,x,c,d>^{10}, <a,x,b,c,d>^{10}, <a,x,b,c,d>^{10}]$ as an example, and Table 1 is used to calculate for activity x . First of all, for activity x , find the activities that have a direct following relationship with x in the relation table, and find that the values of $\#(<x,b>,L)$, $\#(<x,c>,L)$ and $\#(<x,d>,L)$ are all greater than 0, so the set of activities that have a direct following relationship with x is $\{b,c,d\}$, and the number of elements in the set is 3. Secondly, for activity x , find the activities that have a direct preceding relationship with x in the relation table, and find that the values of $\#(<a,x>,L)$, $\#(<b,x>,L)$, $\#(<c,x>,L)$ and $\#(<d,x>,L)$ are all greater than 0, so the set of activities that have a direct following relationship with x is $\{a,b,c,d\}$, and the number of elements in the set is 4. So for activity x ,

the collection of activities that satisfy the conditions is $\{b, c, d\}$ and $\{a, b, c, d\}$. The sum of the number of elements in the two sets is calculated, and the result is 7, so $CH_1(x) = 7$ is obtained. 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 5. For activity x , the calculated result is greater than the average value, so it is judged that the candidate chaotic activities are x .

Condition (2): Chaotic activities in logs often have both direct following relationships and direct preceding relationships with more activities in the log. Accordingly, the amount of activities in the event log that have both a direct following relationship and a direct preceding relationship with a certain activity can be counted to calculate the chaos degree of the activity. That is, for activity p , let the event log be L , and the sum of the number of elements in the set be counted as the degree of chaos in activity p . If the chaos degree of activity p exceeds the set threshold, it is judged that x is a candidate chaotic activity.

$$S_3 = \left\{ q \mid \forall_{q \in ActSet(L)} (\#(< p, q >, L) > 0) \wedge (\#(< q, p >, L) > 0) \right\}$$

Take the log $L = [<a, b, c, d, x>^{10}, <a, b, x, c, d>^{10}, <a, x, b, c, d>^{10}, <a, x, b, c, d>^{10}]$ as an example, and use Table 1 to calculate activity x . First of all, for activity x , find the activities that have both a direct following relationship and direct preceding relationship with x in the relation table, and find 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, and the values of $\#(< x, d >, L)$, $\#(< d, x >, L)$ are all greater than 0. So for activity x , the collection of activities that satisfy the conditions is $\{b, c, d\}$. 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 1.2. The result of activity x is greater than the average, so it is judged that the candidate chaotic activity is x .

Condition (3): If there are both direct following relationships and direct preceding relationships between a certain activity and additional activities, and the difference between the direct following frequency and the direct preceding frequency is small, the activity is judged to be chaotic. The average value of the direct following frequency and the direct preceding frequency between two activities can be set as the threshold to judge the gap between them. Therefore, activities that have both a direct following and direct preceding relationship 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 direct following frequency and the direct preceding frequency between the activities 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 p , let the event log be L , and the sum of the number of elements in the set S_4 be counted as the degree of chaos in activity p . If the chaos degree of activity p exceeds the set threshold, it is judged that p is a candidate chaotic activity.

$$S_4 = \left\{ q \mid \forall_{q \in ActSet(L)} (\#(< p, q >, L) > 0) \wedge (\#(< q, p >, L) > 0) \wedge \left(\left| \#(< p, q >, L) - \#(< q, p >, L) \right| < \frac{\#(< p, q >, L) + \#(< q, p >, L)}{2} \right) \right\}$$

Take the log $L = [<a, b, c, d, x>^{10}, <a, b, x, c, d>^{10}, <a, x, b, c, d>^{10}, <a, x, b, c, d>^{10}]$ as an example, and use Table 1 to calculate activity x . First of all, for activity x , find the activities that have both a direct following relationship and direct preceding relationship with x in the relation table, and find that $\#(< x, b >, L)$, $\#(< b, x >, L)$ are all greater than 0, $|\#(< x, b >, L) - \#(< b, x >, L)| < \frac{\#(< x, b >, L) + \#(< b, x >, L)}{2}$, $\#(< x, c >, L)$, $\#(< c, x >, L)$ are greater than 0, $|\#(< x, c >, L) - \#(< c, x >, L)| < \frac{\#(< x, c >, L) + \#(< c, x >, L)}{2}$, $\#(< x, d >, L)$, $\#(< d, x >, L)$ are all greater than 0, and $|\#(< x, d >, L) - \#(< d, x >, L)| < \frac{\#(< x, d >, L) + \#(< d, x >, L)}{2}$. So for activity x , the collection of activities that satisfies the conditions is $\{b, c, d\}$. 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

1.2. The result of activity x is greater than the average, so it is judged that the candidate chaotic activity is x .

Condition (4): By dividing the value obtained from condition (3) with the value obtained from condition (2), the set of activities that have both a direct following relationship and direct preceding relationship with an activity in the log is obtained through condition (2), and through condition (3), we obtain the set of activities with a small difference between the direct following frequency and the direct preceding frequency between the activities in the log. Calculate the proportion of the latter to the former; this 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.

Take the log $L = [<a,b,c,d,x>^{10}, <a,b,x,c,d>^{10}, <a,x,b,c,d>^{10}, <a,x,b,c,d>^{10}]$ as an example, and use Table 1 to calculate activity x . Firstly, for activity x in the event log, condition (2) is calculated, and $CH_2(x) = 3$ is obtained. For condition (3), $CH_3(x) = 3$, so for condition (4), the calculated result is $CH_4(x) = 1$. This calculation is performed for all of the activities in the log, and the corresponding results of each activity are obtained. For all of the results, the average value is 0.8. 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 $L = [<a,b,c,d,x>^{10}, <a,b,x,c,d>^{10}, <a,x,b,c,d>^{10}, <a,x,b,c,d>^{10}]$ as an example. For condition (1), we obtain $\{CH_1(a) = 2, CH_1(b) = 4, CH_1(c) = 4, CH_1(d) = 3, CH_1(x) = 7\}$, with an average of 5, so the candidate chaotic activities are b , x . For condition (2), it is found that $\{CH_2(a) = 0, CH_2(b) = 1, CH_2(c) = 1, CH_2(d) = 1, CH_2(x) = 3\}$ and the average value is 1.2, so the candidate chaotic activity is x . For condition (3), we find that $\{CH_3(a) = 0, CH_3(b) = 1, CH_3(c) = 1, CH_3(d) = 1, CH_3(x) = 3\}$ and the average value is 1.2, so the candidate chaotic activity is x . For condition (4), it is found that $\{CH_4(a) = 0, CH_4(b) = 1, CH_4(c) = 1, CH_4(d) = 1, CH_4(x) = 1\}$, with an average value of 0.8, and the candidate chaotic activities are b , c , d , and x . Finally, the chaotic activity is x .

3.3. Indirect Chaotic Activity Filtering

An alternative to the approach proposed in Algorithm 1 is to filter out certain activities in the event log to reduce the total chaos degree of the log. Define the overall chaos degree of the event log as the sum of the chaos degree of all activities in the event log, that is, $CH_i(L) = \sum_{y \in ActSet(L)} CH_i(y)$. Algorithm 2 proposes an algorithm that iteratively filters activities from an event log that results in a significant reduction in log chaos, as opposed to Algorithm 1, which selects activities to filter based on the overall clutter of the log after deleting the activity.

By calculating the relationship table among activities corresponding to logs obtained after filtering a certain activity, Algorithm 2 obtains the change degree of the overall chaos degree value of the filtered event log, and identifies the chaotic activities in the event log on the basis of this value. Firstly, the activities 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 significant impact on the overall level of chaos in event logs, and the main purpose of filtering chaotic activities is to improve the quality of process models by reducing the level of chaos in event logs. Consequently, by calculating the change degree of the total log chaos degree after deleting certain activities, the influence degree of the activity on the total log chaos degree 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 $y \in ActSet(L)$ in log L is defined as $CH_i^L(y), CH_i^{L'}(y) = CH_i(L) = CH_i(L \uparrow ActSet(L) \setminus \{y\})$.

The program code is as follows:

Algorithm 2: Indirect Chaotic Activity Filtering

Input: event log L

Output: event log list QLS

```

 $L' \leftarrow L$ 
 $QLS \leftarrow \langle L' \rangle$ 
While  $|ActSet(L')| > 2$ 
   $actset \leftarrow ActSet(L')$ 
   $countos \leftarrow \emptyset, countmos \leftarrow \emptyset, countnos \leftarrow \emptyset$ 
  for  $a$  in  $ActSet(L')$  do
     $L' \leftarrow L' \uparrow actset \setminus \{a\}$ 
     $counts \leftarrow \emptyset, countms \leftarrow \emptyset, countns \leftarrow \emptyset$ 
    for  $i$  in  $ActSet(L')$  do
       $count1 \leftarrow 0, countm \leftarrow 0, countn \leftarrow 0, countmn \leftarrow 0$ 
      for  $j$  in  $ActSet(L')$  do
        if  $(D_{ij}.d_{fs} > 0)$  then
           $count1++$ 
        if  $(D_{ij}.d_{ps} > 0)$  then
           $count1++$ 
        if  $(D_{ij}.d_{fs} > 0 \text{ and } D_{ij}.d_{ps} > 0)$  then
           $countm++$ 
        if  $(|D_{ij}.d_{fs} - D_{ij}.d_{ps}| > 0)$  then
           $countn++$ 
      end for
       $counts \leftarrow \{counts \cup (i, count1)\}$ 
       $countms \leftarrow \{countms \cup (i, countm)\}$ 
       $countns \leftarrow \{countns \cup (i, countn)\}$ 
    end for
     $counto \leftarrow 0, countmo \leftarrow 0, countno \leftarrow 0$ 
    for  $ac$  in  $ActSet(L')$ 
       $counto \leftarrow counto + counts.get(ac)$ 
       $countno \leftarrow countno + countno + countns.get(ac)$ 
    end for
     $countos \leftarrow \{countos \cup (i, counto)\}$ 
     $countmos \leftarrow \{countmos \cup (i, countmo)\}$ 
     $countnos \leftarrow \{countnos \cup (i, countno)\}$ 
  end for
   $rem \leftarrow \emptyset$ 
  for  $a$  in  $ActSet(L')$  do
    if  $(countos.get(a) > average(countos) \text{ and } countmos.get(a) > average(countmos))$  then
      if  $(countnos.get(a) > average(countnos))$  then
         $rem \leftarrow \{rem \cup a\}$ 
      end for
     $L' \leftarrow L' \uparrow actset \setminus rem$ 
     $QLS \leftarrow QLS \cdot \langle L' \rangle$ 
  end while

```

Take $\log L = [\langle a, b, c, d, x \rangle^{10}, \langle a, b, x, c, d \rangle^{10}, \langle a, x, b, c, d \rangle^{10}, \langle a, x, b, c, d \rangle^{10}]$ as an example.

For condition (1), take activity x as an example. After deleting activity x , the event log L' is $L' = [\langle a, b, c, d \rangle^{40}]$. For the calculation of the relation between activities, $\#(\langle a, b \rangle, L') = 40$, $\#(\langle b, c \rangle, L') = 40$, and $\#(\langle c, d \rangle, L') = 40$ is obtained. Thus the corresponding relation tables are obtained, and using the relationship tables to calculate, $\{CH_1(a) = 1, CH_1(b) = 2, CH_1(c) = 2, CH_1(d) = 1\}$ is obtained, so we get $CH_1^L(x) = 6$. This calculation is performed for all activities in the original log, and the results related to each activity are obtained, and the average value of all results is obtained. In this original log, the average value is 12.8. 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 $L' = [<a,c,d,x>^{10}, <a,x,c,d>^{20}, <a,c,x,d>^{10}]$, and the relation between activities is calculated to obtain $\#(<a,c>,L') = 20$, $\#(<a,x>,L') = 20$, $\#(<c,d>,L') = 30$, $\#(<c,x>,L') = 10$, $\#(<d,x>,L') = 10$, $\#(<x,c>,L') = 20$, and $\#(<x,d>,L') = 10$. Thus the corresponding relation tables are obtained, and using the relation tables to calculate, $\{CH_2(a) = 0, CH_2(c) = 1, CH_2(d) = 1, CH_2(x) = 2\}$ is obtained, so we get $CH_2^L(b) = 4$. This calculation is performed for all of the activities in the original log, and the results related to each activity are obtained. The average value of all of the results is obtained. In this original log, the average value is 3.6. 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 $L' = [<b,c,d,x>^{10}, <b,x,c,d>^{10}, <x,b,c,d>^{10}, <b,c,x,d>^{10}]$, and the relation between activities is calculated to obtain $\#(<b,c>,L') = 30$, $\#(<b,x>,L') = 10$, $\#(<c,d>,L') = 30$, $\#(<c,x>,L') = 10$, $\#(<x,b>,L') = 10$, and $\#(<x,c>,L') = 10$, $\#(<x,d>,L') = 10$. Thus, the corresponding relation table is obtained. For the relational table, the calculation for condition (3) gives $\{CH_3(b) = 1, CH_3(c) = 1, CH_3(d) = 1, CH_3(x) = 3\}$. So, we get $CH_3^L(a) = 6$. This calculation is performed for all of the activities in the original log, and the results related to each activity are obtained. The average value of all of the results is obtained. In this original log, the average value is 3.6. 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 condition (1), we obtain $\{CH_1^L(a) = 16, CH_1^L(b) = 14, CH_1^L(c) = 14, CH_1^L(d) = 14, CH_1^L(x) = 6\}$, with an average of 12.8, and so the candidate chaotic activities are x . For condition (2), it is found that $\{CH_2^L(a) = 6, CH_2^L(b) = 4, CH_2^L(c) = 4, CH_2^L(d) = 4, CH_2^L(x) = 0\}$, and the average value is 3.6, so the candidate chaotic activity is x . For condition (3), we find that $\{CH_3^L(a) = 6, CH_3^L(b) = 4, CH_3^L(c) = 4, CH_3^L(d) = 4, CH_3^L(x) = 0\}$, and the average value is 3.6, so the candidate chaotic activity is x . Finally, the chaotic activity is x .

4. Results

This section conducts comparative experiments on different chaotic activity filtering methods using simulated event logs and real event logs, and the proposed chaotic activity filtering approaches were evaluated experimentally. Table 2 shows some main statistical data of real event logs. Tables 3–5 show the results of experiments using simulation logs. Figure 2 shows the direct following graph of the real event log, Figures 3 and 4 show the direct following graphs of the filtered event logs obtained by using the proposed chaotic activity filtering methods, and Figures 5–7 show the results obtained from experiments using real event logs. The experimental results demonstrate that the filtering method proposed in this paper has certain advantages, that is, it can improve the accuracy and running speed of chaotic activity filtering.

Table 2. Real event logs used in the experiment.

Data Set	Track Number	Number of Events	Activity Number
Environmental Permit	1434	8577	27
Sepsis	1050	15,214	16
BPI Challenge 2012	13,087	164,506	23

Table 3. Comparison of misdeletion rate of different filtering approach (Logs contain k frequent randomly-positioned activities) (%) (ms).

Approach	frequent(1)		frequent(2)		frequent(3)		frequent(4)		frequent(5)		frequent(6)		frequent(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($\alpha = 1/ A $)	0	53	0	57	0	58	0	63	0	70	0	79	0	96
Indirect	0	89	0	105	0	114	0	126	0	138	0	156	0	168
Indirect($\alpha = 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 4. Comparison of misdeletion rate of different filtering approach (Logs contain k uniform randomly-positioned activities) (ms).

Approach	random(1)		random(2)		random(3)		random(4)		random(5)		random(6)		random(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($\alpha = 1/ A $)	0	51	0	54	0	56	0	60	0	66	0	75	0	82
Indirect	0.143	82	0	95	0	107	0	129	0	136	0	138	0	158
Indirect($\alpha = 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 5. Comparison of misdeletion rate of different filtering approach (Logs contain k infrequent randomly-positioned activities) (ms).

Approach	infrequent(1)		infrequent(2)		infrequent(3)		infrequent(4)		infrequent(5)		infrequent(6)		infrequent(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($\alpha = 1/ A $)	0	46	0	49	0	51	0	56	0	60	0	67	0	72
Indirect	0.143	79	0	90	0.143	104	0.143	111	0.143	121	0.143	126	0.143	141
Indirect($\alpha = 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	1	98	0	116

4.1. Data Set and Experimental Setup

This section firstly introduces the method of evaluating the effectiveness of the chaotic activity filtering method by using the simulation event log containing chaotic activity, and then introduces the method of evaluating the effectiveness and efficiency of the chaotic activity filtering method by using the real event log, as well as the relevant information of the real event log used in the experiment.

4.1.1. Simulation Log

Firstly, by simulating event logs containing chaotic activities, the event logs before and after chaotic activity filtering can be compared to determine whether the filtering method can identify all chaotic activities in the logs and calculate the normal number of activities that need to be deleted from the logs if all randomly inserted activities are filtered out.

In order to verify the accuracy of the chaotic activity filtering approach, first, in **step (1)**, a synthetic event log is generated from the process model to determine that there is no chaotic activity in the event log. Then, in **step (2)**, activities are manually inserted at random positions in the log. Due to the locations in these activity logs being randomly selected, it is assumed that these activities are chaotic. Change the number (k) of random positioning activities inserted to evaluate how chaotic activity filtering methods

can handle different amounts of random positioning activities in event logs. In addition, the frequency of inserted random positioning activities is changed, among which three types of random positioning activities are distinguished. Firstly, the frequency of occurrence for each activity in the event log is calculated and then different random positioning activity insertion methods are used to insert chaotic activities with different frequencies in the log: ① *frequent(k)*: Frequent random positioning activities are inserted into the log, and frequent random positioning activities indicate that in the log, the frequency of occurrence for each activity out of k random positioning activities is the highest frequency of occurrence for the activities in the event log. ② *infrequent(k)*: Infrequent random positioning activities are inserted into the log, and the infrequent random positioning activity indicates that in the log, the frequency of each activity in the k random positioning activities is the frequency of the activity with the lowest occurrence frequency in the event log. ③ *random(k)*: Evenly random positioning activities are inserted into the log, and for each of the k random positioning activities, the frequency of each activity is obtained by taking any value between the frequency of the activity with the highest frequency and the frequency of the activity with the lowest frequency in the event log. In **step (3)**, different chaotic activity filtering methods are used to filter logs with different quantities and frequencies, and all inserted random positioning activities are deleted until all k random positioning activities are deleted. Then, calculate the number (k) of initial activities in the process model that were deleted from the log during this process, calculate the false deletion rate of the filtering method, that is, the amount of false deleted activities/the amount of original activities, and calculate the time required to delete k manually inserted activities (**step (4)**).

4.1.2. Real Log

Secondly, in the evaluation of real data, indirect evaluation methods are needed because there is no relevant information about chaotic activities in the logs, that is, after using different chaotic activity filtering techniques to filter out chaotic activities in the event log, we evaluated the quality of the filtered process model discovered from the filtered event log and the time required to complete the chaotic activity filtering of the event log. An experimental evaluation of the chaotic activity filtering approaches proposed in this article used three real event logs. Table 2 shows some main statistical data of these event logs.

Environmental Permit: The data set 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 1000 cases, and a total of 15,000 events recorded across 16 different activities.

BPI Challenge 2012: This event log involves the loan application process of Dutch financial institutions. The cases in the log contain the main application information and the objection procedures at each stage.

4.2. Evaluation Indicators

This section introduces the evaluation indicators used to evaluate the effectiveness and efficiency of the chaotic activity filtering method by using a simulation event log and real event log, respectively.

4.2.1. Simulation Log

For synthesizing event logs, calculate the error deletion rates of different chaotic activity filtering methods, and evaluate the quality of chaotic activity filtering methods through these methods. If this filtering approach is used to filter out all of 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 preprocess the event log to improve the quality of the filtered process model. Therefore, to evaluate the accuracy of the chaotic activity filtering method, the quality of the process model obtained by the chaotic activity filtering method can be evaluated. Firstly, different chaotic activity filtering techniques are used to filter the real event log to obtain the filtered event log. Then, the filtered process model is obtained by using the process discovery algorithm for the filtered log. Evaluate the effectiveness of the chaotic activity filtering method by evaluating the quality of the obtained process model.

In real data valuation, due to the absence of relevant information on chaotic activities in the log, a more indirect evaluation is needed. After using the direct and indirect filtering methods mentioned above, as well as the filtering method proposed in reference [2], to filter out the chaotic activities identified in the event log, we evaluate the quality of the filtered process model obtained through the process discovery algorithm. In this section, we evaluate the effectiveness of chaotic activity filtering techniques by evaluating the quality of the discovered process models. The efficiency of the filtering method is evaluated by calculating the running time of the filtering method when filtering activity in each log.

This section evaluates the quality of the process model for the event log using a quantitative method for calculating the F-score value. The IM algorithm was used to mine event logs to obtain a model. For each filtered process model we discovered, the *fitness* and *precision* [21] of the filtered process model were measured. *Fitness* is measured using an alignment-based fitness measurement [22], and we measure *precision* using negative event precision [23]. Then, using the results of *fitness* and *precision* obtained by calculation, we calculated the *F-score* [24].

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

The direct filtering approach and indirect 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

Firstly, this section shows and analyzes the experimental results obtained by comparing the traditional filtering method with the chaotic activity filtering method proposed in this paper on different types of simulation data sets. Then the experimental results obtained by comparing the traditional filtering method with the chaotic activity filtering method proposed in this paper on different real data sets are shown and analyzed.

4.3.1. Simulation Log

The number of chaotic activities in the event log and the frequency of chaotic activities occurring in the log are uncertain. Therefore, inserting random positioning activities into the log can be used to simulate event logs containing chaotic activities. Random positioning activities are inserted randomly in the log, and random positioning activities with different frequencies can be obtained through different insertion methods.

First, an experiment was conducted on the synthetic event log with k random positioning activities inserted, and the results were obtained as follows. Table 3 shows the experimental results obtained by inserting frequent random positioning activities into the log. The method *frequent(k)* was used to insert k frequent random positioning activities into the log. The frequency of the k random positioning activities is the frequency of the activity with the highest frequency in the log. Table 4 is the experimental result of inserting uniform random positioning activities into the log. The method *random(k)* of inserting uniform random positioning activities is used to insert k uniform random positioning activities into the log. For each of the k inserted random positioning activities, the frequency of occurrence is randomly selected from the uniform probability distribution. Table 5 shows

the experimental results obtained by inserting infrequent random positioning activities into the log. The infrequent random positioning activity insertion method *infrequent(k)* is used to insert k infrequent random positioning activities into the log. The occurrence frequency of k random positioning activities is the occurrence frequency of the activity with the lowest occurrence frequency in the log. The experimental results show the error deletion rate generated in the process of using different filtering methods to filter chaotic activities in event logs, that is, the number of false deletion activities/the number of original activities, which is used to evaluate the effect of the filtering method. If the error deletion rate is high, it indicates that the method has a low accuracy in identifying chaotic activities.

In the experiment, the Direct(dfr) and Indirect(dfr) approaches proposed in this paper were compared with the four methods proposed in reference [2] (direct entropy-based activity filtering approach Direct, direct entropy-based activity filtering approach with Laplace smoothing $\text{Direct}(\alpha = 1/|A|)$, indirect entropy-based activity filtering approach Indirect, and indirect entropy-based activity filtering approach with Laplace smoothing $\text{Indirect}(\alpha = 1/|A|)$) to compare the running time and error deletion rate of the approaches, that is, the accuracy of the filtering approaches.

According to the deletion error rate generated by different filtering methods in the process of deleting all chaotic activities in logs, as shown in Table 3 to Table 5, firstly, infrequent random positioning activities have a greater impact on the accuracy of the filtering method for chaotic activities than frequent random positioning activities. Both the number and frequency of random location activities have a certain influence on the effect of the chaotic activity filtering algorithm. Secondly, the chaotic activity filtering method proposed in this paper can accurately distinguish the normal activity from the chaotic activity in the log, and the accuracy of the indirect filtering method proposed in this paper is better than that of the entropy-based indirect filtering method, and the error deletion rate of the chaotic activity filtering method proposed in this paper is basically the same as that of the direct filtering method based on entropy. Therefore, it can be concluded that although the filtering methods used in the experiment have different accuracy rates for event logs with different frequencies and quantities of random positioning activities, the chaotic activity filtering method proposed in this paper can effectively improve the accuracy of chaotic activity filtering compared with other chaotic activity filtering methods.

The following conclusions can be drawn from the running time required by different filtering methods shown in Tables 3–5 to complete the filtering of chaotic activities. First, with the increase in k value, the running time required to filter all chaotic activities in the log increases. Meanwhile, frequent random positioning activities have a greater impact on the running time than infrequent random positioning activities. Therefore, both the number and frequency of random positioning activities have a certain influence on the effect of the chaotic activity filtering algorithm, and the number and frequency of chaotic activities are proportional to the running time of the filtering method. Secondly, the direct filtration method proposed in this paper has a shorter running time and higher efficiency than the entropy-based filtration method. Moreover, compared with the entropy-based indirect chaotic activity filtering method, the proposed indirect chaotic activity filtering method has a higher operating efficiency.

Therefore, by using the entropy based chaotic activity filtering method and the direct chaotic activity filtering method proposed in this chapter, it can be concluded that although the filtering method used in the experiment presents a different time performance for event logs with different frequencies and quantities of random activities, that is, the running time difference between the proposed filtering method and the traditional filtering method is different, compared with other chaotic activity filtering methods, the proposed method can improve the running efficiency of the chaotic activity filtering algorithm.

In Table 3 to Table 5, the filtration methods proposed in this paper are Direct(dfr) and Indirect(dfr). The methods used for comparison experiments with the proposed method are the direct entropy-based activity filtering approach (Direct), direct entropy-based activity filtering approach with Laplace smoothing ($\text{Direct}(\alpha = 1/|A|)$), indirect entropy-based

activity filtering approach (Indirect), and indirect entropy-based activity filtering approach with Laplace smoothing ($\text{Indirect}(\alpha = 1/|A|)$).

4.3.2. Real Log

Using real event logs for experiments, compare the accuracy and running time of the Direct(dfr) and Indirect(dfr) methods discussed in this paper with the four methods proposed in reference [1]: (Direct, $\text{Direct}(\alpha = 1/|A|)$, Indirect and $\text{Indirect}(\alpha = 1/|A|)$), and the least-frequent-first chaotic activity filtering approach Infrequent. In experiments with real event logs, the quality of chaotic activity filtering technology was evaluated by an indirect method, which evaluated the quality of the model obtained after the process mining of the event logs obtained after chaotic activity filtering had been completed. The model evaluation method used in this section is to calculate the harmonic mean between the *fitness* and *precision* of the model.

An intuitive presentation of the effectiveness of the chaotic activity filtering method is obtained by observing the direct following graph corresponding to the event log. Taking event log Sepsis as an example, Figure 2 shows the direct following graph of Sepsis. Based on the direct following graph of logs, it can be seen that the direct following relationship between activities in event logs is chaotic. Figure 3 shows the direct following graph corresponding to all filtered event logs obtained by filtering the logs using the direct chaotic activity filtering method proposed in this chapter.

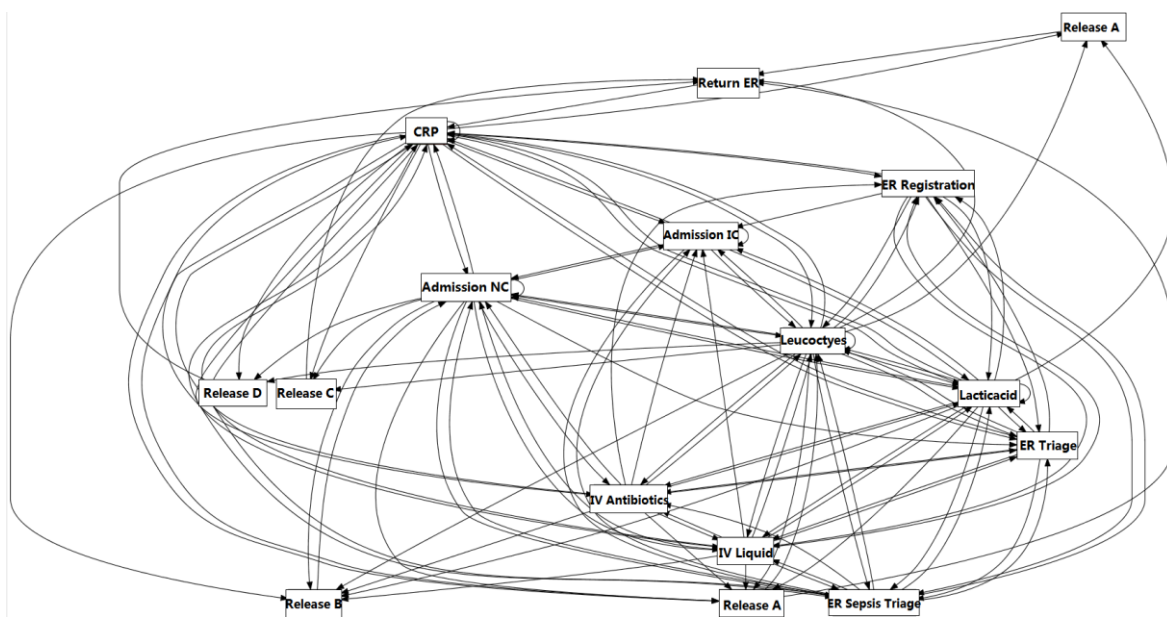


Figure 2. Direct following graph for Sepsis.

According to Figure 3, the event log Sepsis is filtered by the direct chaotic activity filtering approaches proposed in this paper to obtain three filtered logs, and the number of remaining activities in the logs is nine and seven in turn. Figure 4 shows the direct following graph corresponding to all filtered event logs obtained by filtering the logs using the indirect chaotic activity filtering method proposed in this chapter. In Figure 4, the event log Sepsis is filtered by the indirect chaotic activity filtering approaches proposed in this paper to obtain three filtered logs, and the number of remaining activities in the logs is eight and seven in turn.

Multiple filtered event logs can be obtained from original event logs after chaotic activity filtering. Figure 2 to Figure 4 show the direct following graphs of all filtered event logs obtained from real event logs after filtering. It can be seen from the direct following graph that after the filtering of chaotic activities, the chaos degree of the direct following

relationship between activities in the log is significantly reduced. As the number of activities decreases, the confusion of the direct following relationship between activities decreases. Moreover, according to the conditions proposed in this paper for the identification of chaotic activities, the chaotic activity filtering method based on the relationship between activities can stop filtering when the number of remaining activities reaches a certain degree, which is related to the log.

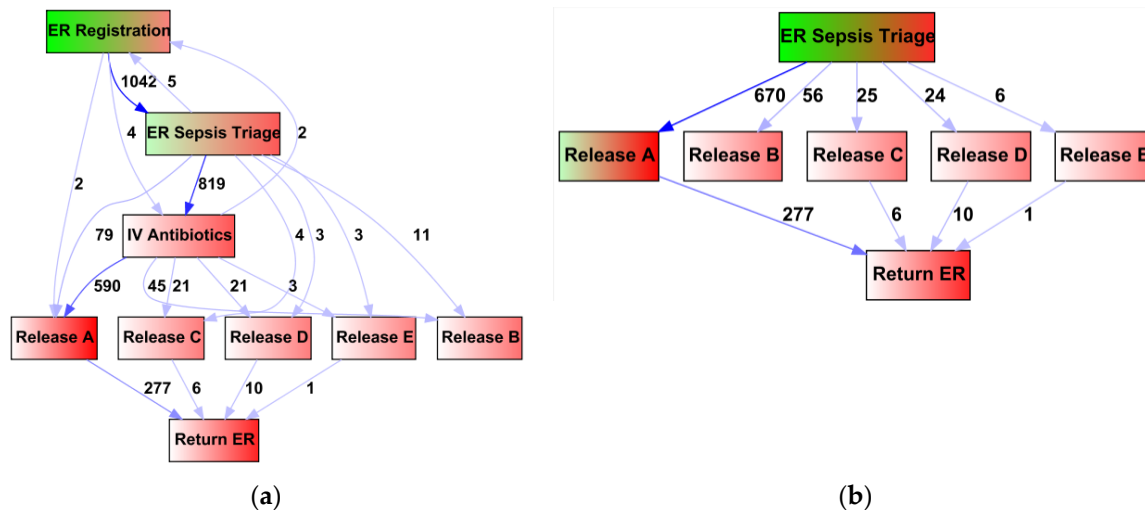


Figure 3. Direct following graph for Sepsis-direct. (a) Direct following graph of filtered event log containing nine activities. (b) Direct following graph of filtered event log containing seven activities.

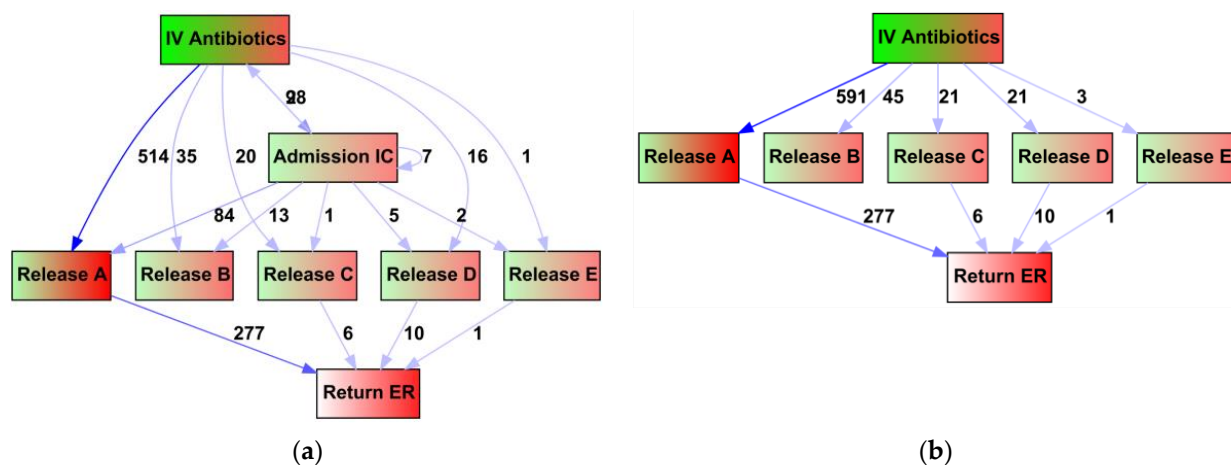


Figure 4. Direct following graph for Sepsis-indirect. (a) Direct following graph of filtered event log containing eight activities. (b) Direct following graph of filtered event log containing seven activities.

As can be seen from the figure, the filtering methods proposed above can filter multiple chaotic activities in each run, so the algorithm can improve the efficiency of the chaotic activity filtering approach. These methods can control the amount of remaining activities in the filtered log within a reasonable range, so as to ensure the integrity of the event log to a certain extent. After filtering the chaotic activities in the event log, by observing the direct following graph of the filtered log, it can be found that the chaotic degree of the relationship between activities in the log is effectively reduced. Therefore, these methods can effectively reduce the chaotic degree of the event log and improve the quality of the model.

The results of the experiment are revealed in Figures 5–7, and each picture consists of two parts. Part (a) describes the contrast of model quality changes with the reduction of the amount of remaining activities in the event log after filtering, in other words, the

accuracy comparison of filtering approaches. Part (b) describes the contrast of the time required to remove all chaotic activity from the event log. Figure 5 shows the comparison of the effectiveness and efficiency between the traditional filtering method and the chaotic activity filtering method put forward in this article for BPI Challenge 2012, and Figure 6 shows the comparison of the effectiveness and efficiency between the traditional filtering method and the chaotic activity filtering method put forward in this article for Sepsis. Figure 7 shows the comparison of the effectiveness and efficiency between the traditional filtering method and the chaotic activity filtering method put forward in this article for Environmental Permit.

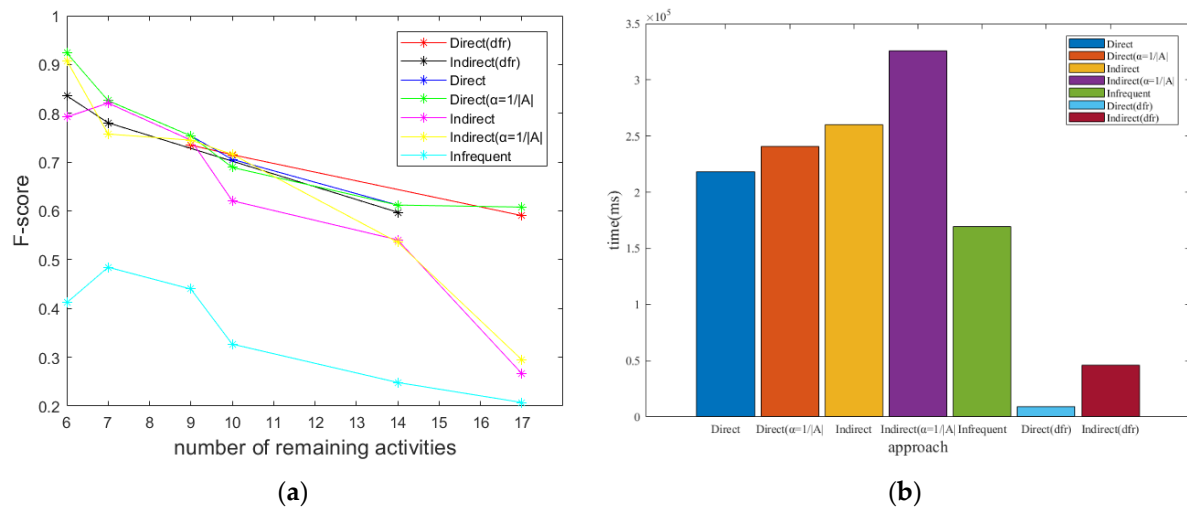


Figure 5. Experimental result for BPI Challenge 2012. (a) Model quality comparison. (b) running time comparison.

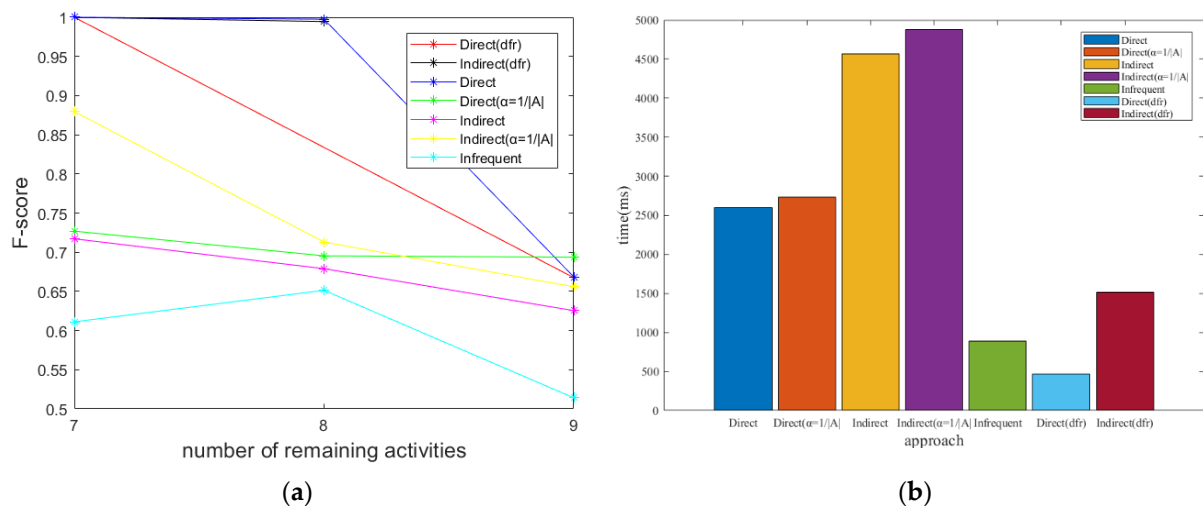


Figure 6. Experimental result of Sepsis. (a) Model quality comparison. (b) running time comparison.

Firstly, different chaotic activity filtering methods were used to filter the event log, and then the process models were obtained by using the process discovery algorithm after filtering the event log, and the quality of the process model was evaluated, so as to assess the accuracy of the chaotic activity filtering method.

In Figure 5 to Figure 7, the filtering methods proposed in this paper are Direct(dfr) and Indirect(dfr). The methods used for comparison experiments with the proposed method are the direct entropy-based activity filtering approach (Direct), direct entropy-based activity filtering approach with Laplace smoothing (Direct($\alpha = 1/|A|$)), indirect entropy-based

activity filtering approach (Indirect), indirect entropy-based activity filtering approach with Laplace smoothing ($\text{Indirect}(\alpha = 1/|A|)$), and the least-frequent-first chaotic activity filtering approach Infrequent.

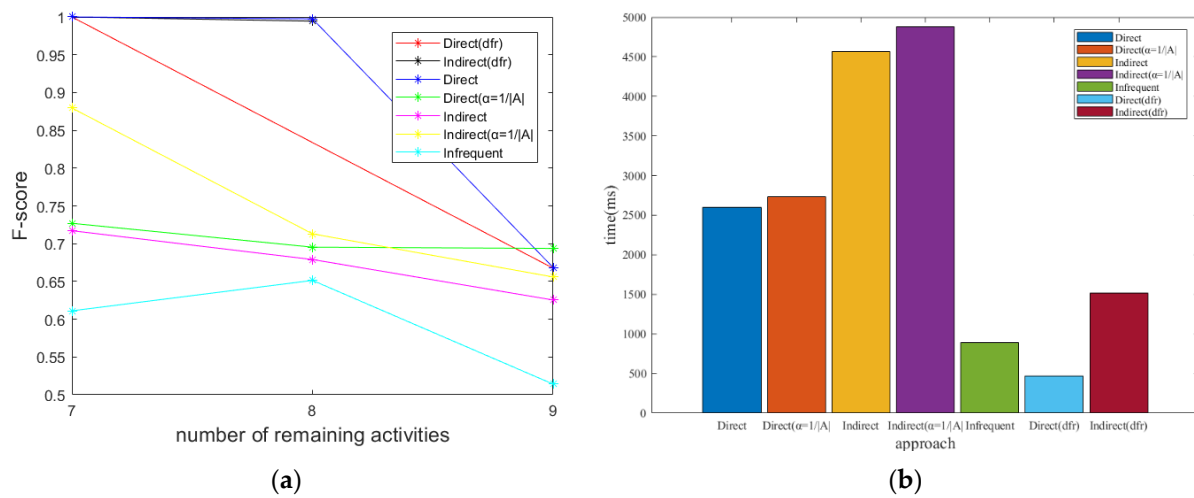


Figure 7. Experimental result of Environmental Permit. (a) Model quality comparison. (b) running time comparison.

The following conclusions can be drawn by observing the accuracy comparison results of the following chaotic activity filtering methods. From the experimental results in Figure 5a to Figure 7a, it can be concluded that, first of all, the quality of the process model is gradually improved as the number of activities in the event log decreases. Moreover, the accuracy of the chaotic activity filtering method is closely related to the event log itself, that is, for different event logs, the accuracy comparison results presented by the chaotic activity filtering method are different. Secondly, the chaotic activity filtering method proposed in this paper is superior to the frequency-based chaotic activity filtering method in terms of accuracy. This method has certain advantages in the accuracy of chaotic activity filtering, and can maintain the integrity of the filtered event log to a certain extent. The traditional chaotic activity filtering method deletes the identified chaotic activities from the log successively until there are two activities in the log, so it is difficult to ensure the integrity of the filtered log. However, the methods proposed in this paper identify and filter 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. Finally, it can be observed that in some event logs, compared with the traditional chaotic activity filtering method, the accuracy curve of the process model after filtering can reach a stable state faster. Therefore, the filtering method 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.

According to the experimental results shown in Figure 5b to Figure 7b, the following conclusions can be drawn: First, for different real event logs, the running time of the chaotic activity filtering method is related to the number of activities in the log and the occurrence frequency of activities. Moreover, the running time of the Indirect filtering method is higher than that of the Direct filtering method, and the running time of the filtering method with Laplace smoothing ($\text{Direct}(\alpha = 1/|A|)$, $\text{Indirect}(\alpha = 1/|A|)$) is higher than that of the filtering method without Laplace smoothing (direct, indirect). Secondly, through

observation, it can be seen that the running time of the direct chaotic activity filtering method proposed in this paper is significantly less than that of the traditional frequency-based chaotic activity filtering method and the four chaotic activity filtering methods based on activity entropy. The indirect chaotic activity filtering method proposed in this paper is significantly superior to the indirect chaotic activity filtering method based on activity entropy in terms of operational efficiency. Therefore, through comparative experiments, it can be concluded that although the difference in time performance between different chaotic activity filtering methods for different event logs is related to the characteristics of event logs, the chaotic activity filtering method proposed in this paper is compared with the traditional chaotic activity filtering method. In terms of time performance, the chaotic activity filtering method proposed in this paper can effectively improve the efficiency of chaotic activity filtering because of the traditional chaotic activity filtering algorithm.

The results of experiment show 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 a certain integrity of the log.

5. Conclusions

In this article, approaches to filtering chaotic activities by using the relationship between activities have been proposed. The approaches can identify and filter chaotic activities by extracting the direct following relationships and direct preceding relationships between activities in the event log. These approaches can identify and delete multiple activities in each algorithm run by using the judgment conditions, and these methods do not involve complex operations. The proposed methods are compared with the frequency-based chaotic activity filtering method and the entropy-based chaotic activity filtering methods by using simulation logs and real event logs. It can be seen that the accuracy and time performance of the proposed filtering method are closely related to the used log itself, that is, for different event logs, the filtering method proposed in this paper presents different effects. However, it can be observed that the accuracy of the proposed filtering methods is better than that of the frequency-based filtering method, and is close to that of the entropy-based chaotic activity filtering method, so the proposed methods have certain advantages in the accuracy of chaotic activity filtering. Moreover, the proposed filtering methods are superior to the traditional frequency-based filtering method and the entropy-based filtering methods in terms of time performance. For simulation event logs, the filtering methods proposed in this paper can improve the operating efficiency by 23.4% on average, and for real event logs, the filtering methods proposed in this paper can improve the operating efficiency by 84.25% on average. It can be concluded that the filtering methods proposed in this paper can effectively shorten the running time required for chaotic activity filtering and effectively improve the efficiency of chaotic filtering. Therefore, the methods proposed in this paper can reduce the pre-processing time of the event log and improve the efficiency without losing too much accuracy. In general, these methods can ensure that the filtered event log has a certain integrity. Meanwhile, since the current mainstream method for the indirect evaluation of the accuracy of the filtering method is to evaluate the quality of the filtered process model obtained after filtering the event log, the smaller the number of filtered event logs obtained after filtering the original event log, the shorter the time required for the evaluation of the filtering method. The filtering approaches proposed in this paper can also reduce the number of logs obtained after filtering the original logs, so they can shorten the time required for the indirect evaluation of chaotic activity filtering approaches. By using a simulation log and real event log to test, it can be concluded that the direct filtering method and indirect method based on the relationship between activities put forward in this article have certain effectiveness and high efficiency.

6. Limitations and Future Works

There is a partial loss of accuracy in identifying chaotic activities. Moreover, the effectiveness of the algorithm is strongly correlated with the event logs themselves. 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.

Author Contributions: Conceptualization, T.L. and N.G.; methodology, T.L.; software, T.L.; investigation, K.L.; writing—original draft preparation, T.L.; writing—review and editing, N.G. and X.G.; supervision, X.G.; funding acquisition, X.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by the Shandong Provincial Undergraduate Teaching Reform Project (Grant Number: Z2021450), National College Students' Innovation and Entrepreneurship Training Program (Grant Number: 202310433069), the Shandong Provincial Natural Science Foundation of P.R. China (Grant Number: ZR2020QF06), and the Shandong University of Technology Postgraduate Teaching Reform Project (Grant Number: 4053222063).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data utilized in this manuscript will be made available on reasonable request.

Acknowledgments: The authors would like to thank the editor and the anonymous reviewers for their constructive comments and suggestions to improve the quality of this paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Reinkemeyer, L. (Ed.) *Process Mining in Action: Principles, Use Cases and Outlook*; Springer International Publishing: Cham, Switzerland, 2020.
2. Tax, N.; Sidorova, N.; van der Aalst, W.M. Discovering more precise process models from event logs by filtering out chaotic activities. *J. Intell. Inf. Syst.* **2019**, *52*, 107–139. [\[CrossRef\]](#)
3. Conforti, R.; Rosa, M.L.; Ter Hofstede, A.H.M. Filtering Out Infrequent Behaviour from Business Process Event Logs. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 300–314. [\[CrossRef\]](#)
4. Lu, X.; Fahland, D.; van den Biggelaar, F.J.H.M.; van der Aalst, W.M.P. Detecting deviating behaviors without models. In Proceedings of the International Workshop on Business Process Intelligence, Innsbruck, Austria, 31 August–3 September 2015; pp. 126–139.
5. Sani, M.F.; van Zelst, S.J.; van der Aalst, W.M.P. Improving process discovery results by filtering outliers using conditional behavioural probabilities. In Proceedings of the International Workshop on Business Process Intelligence, Barcelona, Spain, 10–11 September 2018; pp. 216–229.
6. van Zelst, S.J.; Sani, M.F.; Ostovar, A.; Conforti, R.; La Rosa, M. Detection and removal of infrequent behavior from event streams of business processes. *Inf. Syst.* **2020**, *90*, 101451. [\[CrossRef\]](#)
7. Smiti, A. A critical overview of outlier detection methods. *Comput. Sci. Rev.* **2020**, *38*, 100306. [\[CrossRef\]](#)
8. Grisold, T.; Wurm, B.; Mendling, J. Using Process Mining to Support Theorizing About Change in Organizations. In Proceedings of the 53rd Hawaii International Conference on System Sciences, Maui, HI, USA, 7–10 January 2020.
9. Leemans SJ, J.; Fahland, D.; van der Aalst, W.M.P. Discovering Block-Structured Process Models from Event Logs Containing Infrequent Behaviour. In Proceedings of the International Conference on Business Process Management, Beijing, China, 26–30 August 2013; pp. 66–78.
10. Guo, C.; Wang, B.; Wu, Z.; Ren, M.; He, Y.; Albarracín, R.; Dong, M. Transformer failure diagnosis using fuzzy association rule mining combined with case-based reasoning. *IET Gener. Transm. Distrib.* **2020**, *14*, 2202–2208. [\[CrossRef\]](#)
11. Benevento, E.; Aloini, D.; van der Aalst, W.M.P. How Can Interactive Process Discovery Address Data Quality Issues in Real Business Settings? Evidence from a Case Study in Healthcare. *J. Biomed. Inform.* **2022**, *130*, 104083. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Ghionnal, L.; Greco, G.; Guzzo, A.; Pontieri, L. Outliner Detection Techniques for Process Mining Application. In Proceedings of the Foundations of Intelligent Systems, International Symposium, Ismi 2008, Toronto, ON, Canada, 20–23 May 2008; Proceedings; DBLP; pp. 150–159.
13. Cheng, H.J.; Kumar, A. Process mining on noisy logs—Can log sanitization help to improve performance? *Decis. Support Syst.* **2015**, *79*, 138–149. [\[CrossRef\]](#)

14. Koschmider, A.; Kaczmarek, K.; Krause, M.; van Zelst, S.J. Demystifying Noise and Outliers in Event Logs: Review and Future Directions. In *Business Process Management Workshops, Münster, Germany, 11–16 September 2022*; Marrella, A., Weber, B., Eds.; Lecture Notes in Business Information Processing; Springer International Publishing: Cham, Switzerland, 2022; pp. 123–135.
15. Guo, Y. Research on Model Discovery and Repair Methods for Process Mining. Master's Thesis, Shandong University of Science and Technology, Qingdao, China, 2018.
16. Leemans, S.J.J.; Fahland, D.; Van Der Aalst, W.M.P. Process and Deviation Exploration with Inductive Visual Miner. In *Proceedings of the BPM (Demos) 2014, Eindhoven, The Netherlands, 10 September 2014*; p. 1295.
17. Lamghari, Z.; Saidi, R.; Radgui, M.; Rahmani, M.D. Chaotic activities recognising during the pre-processing event data phase. *Int. J. Bus. Intell. Data Min.* **2022**, *20*, 412–439. [[CrossRef](#)]
18. Lu, K.; Fang, X.; Fang, N.; Asare, E. Discovery of effective infrequent sequences based on maximum probability path. *Connect. Sci.* **2022**, *34*, 63–82. [[CrossRef](#)]
19. Li, J.; Fang, X.; Zuo, Y. Entropy-Based Behavioral Closeness Filtering Chaotic Activity Method. *Mathematics* **2024**, *12*, 666. [[CrossRef](#)]
20. de Leoni, M.; Pellattiero, L. The benefits of sensor-measurement aggregation in discovering IoT process models: A smart-house case study. In *International Conference on Business Process Management*; Springer International Publishing: Cham, Switzerland, 2021; pp. 403–415.
21. Marin-Castro, H.M.; Tello-Leal, E. An end-to-end approach and tool for BPMN process discovery. *Expert Syst. Appl.* **2021**, *174*, 114662. [[CrossRef](#)]
22. Adriansyah, A.; van Dongen, B.F.; van der Aalst, W.M.P. Conformance checking using cost-based fitness analysis. In *Proceedings of the 15 IEEE International Enterprise Distributed Object Computing Conference (EDOC), Helsinki, Finland, 29 August–2 September 2011*; IEEE: Piscataway, NJ, USA, 2011; pp. 55–64.
23. vanden Broucke, S.K.L.M.; De Weerd, J.; Vanthienen, J.; Baesens, B. Determining process model precision and generalization with weighted artificial negative events. *IEEE Trans. Knowl. Data Eng.* **2013**, *26*, 1877–1889. [[CrossRef](#)]
24. De Weerd, J.; De Backer, M.; Vanthienen, J.; Baesens, B. A robust F-measure for evaluating discovered process models. In *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM), Paris, France, 11–15 April 2011*; IEEE: Piscataway, NJ, USA, 2011; pp. 148–155.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.