IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 29, 2019, accepted August 23, 2019, date of publication September 6, 2019, date of current version September 25, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2939565

A Self-Adaptive Process Mining Algorithm Based on Information Entropy to Deal With Uncertain Data

WEIMIN LI[1], (Member, IEEE), YUTING FAN[1], WEI LIU[1], MINJUN XIN[1], HAO WANG[2], (Member, IEEE), AND QUN JIN[3], (Member, IEEE)

[1]School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China
[2]Department of ICT and Natural Sciences, Norwegian University of Science and Technology, 6025 Aalesund, Norway
[3]Faculty of Human Sciences, Waseda University, Tokorozawa 359-1192, Japan

Corresponding author: Weimin Li (wmli@shu.edu.cn)

This work was supported by the National Key R&D Program of China (No. 2017YFE0117500), and in part by the Natural Science Foundation of China under Grant 61762002.

ABSTRACT Process mining is a technology to gain knowledge of the business process by using the event logs and achieve a model of the process, which contributes to the detection and improvement of the business process. However, most existing process mining algorithms have drawbacks associated with managing uncertain data, and the method of using the frequency threshold alone needs to be enhanced. This paper improves correlation measures in heuristic mining to build a correlation matrix based on an improved frequency matrix. Combined with the maximum entropy principle, a self-adaptive method to determine the threshold is given, which is used to remove the uncertain data relationship in the logs. Furthermore, this study identifies a selective and parallel structure through a modified frequency matrix, and we can get a Petri net-based process model from a directed graph. The recognition of parallel structures contributes to eliminating imbalances when calculating the threshold to deal with the uncertain data. Finally, this paper presents an algorithm framework for adaptively removing uncertain data. This study represents a new attempt to use entropy to remove uncertain data in the field of Business Process Management (BPM). The threshold to deal with the uncertain data does not need to set the parameters in advance. Therefore, the proposed algorithm is self-adaptive and universal. Experimental results show that the algorithm proposed in this study has a higher degree of behavioral and structural appropriateness, and fitness, for the uncertain log data compared to traditional algorithms.

INDEX TERMS Process mining, entropy, maximum entropy principle, uncertain data, BPM.

I. INTRODUCTION

With the development of informatization and the increasingly intense competition among enterprises, it is necessary to optimize the workflow of the enterprise to improve work efficiency. Many enterprises have begun to use various business process management systems to manage enterprise business processes and to automate and intelligent enterprise processes. Most business process management systems not only manage the activities contained in the business process, but also record the sequence of events that are triggered or processed during the execution of the business process, and automatically generate an event log. These logs are a record of actual business execution. The event log contains a lot of information, such as a timestamp indicating the start and end time of the event, operator information of the event, resource information consumed by the event, and so on. Data mining in workflow management is to mine the log data, especially the uncertain trace data, find out the actual running mode of the workflow in order to reproduce the real process of the business process, analyze and optimize the workflow.

Although many mining algorithms that can extract workflow models from event logs are proposed in business process management, and some common structures such as

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

loops, repetitive tasks, non-free choice structures, invisible tasks, and uncertain trajectories are processed, there are still some problems with the data and so on. Medeiros et al. proposed an $\alpha+$ process mining algorithm to process short loops [1], [2]. Wen et al. used a Petri-net description to study non-free choice structures through causal dependencies between tasks [3]. In addition, they proposed $\alpha\#$ algorithm that extends the mining ability of classical α algorithm to complete the mining of invisible tasks [4]. Weijters et al. designed a heuristic algorithm for process mining that can solve some uncertain data, but it has lower adaptability [5]. The subjective threshold of the user does not correctly eliminate the indeterminate data, and may also result in the removal of the correct arc. The heuristic algorithm [6], [18] considers the frequency of event log, but it has bad versatility. In this paper, we consider uncertain data in business process, and aim to solve the impact coming from uncertain data. So the proposed method can ensure the accuracy of the mining results.

Uncertain trace data refers to errors or atypical trajectories appeared in the log data, such as trajectories that are incorrectly saved during business process execution, or atypical process model behavior. If process mining does not analyze and differentiate the uncertain trajectory data, the traditional process mining algorithm will form a process model that does not match to the actual process. These algorithms assume that there is no uncertain trajectories data in the log; the problem of process mining is explored in this situation. Since it is really difficult to guarantee that these hypothesis are consistent with the correct event log, it is important to treat the uncertain trajectory data.

In order to solve the problem of threshold selection while removing uncertain data, this paper proposes a processing of uncertain data of process mining based on information entropy. This method can treat the problem of uncertain data and gain the correct routing structure. In this study, information entropy is added to process mining, and the threshold of processing of uncertain data is adaptively determined according to the principle of maximum entropy. The rationality of integrating information entropy with process mining is verified in this study. In addition, this paper defines the selective and parallel structure based on the frequency of activity dependency in the log and provides a way to transform the directed graph into the process model based on Petri nets.

The remainder of this paper is structured as follows. Related work on process mining is introduced in Section 2. Section 3 provides the relevant definition and treatment of uncertain log data. Section 4 outlines the overall framework of the algorithm. Section 5 reveals the results and analysis of the experiments. Finally, we give the conclusion of this study.

II. RELATED WORKS

The concept of process mining was proposed in 1995 by Professor Cook and Wolf [7], in which the goal of discovering process models based on the given log data automatically. It is introduced into the domain of business process by Agrawal *et al.* [8], and the authors used a directed non-cyclic graph to model the process model. However, it did not represent concurrency well. Israel's IBM's Pinter et al. used the beginning and ending of events in the log data to distinguish events so that time series can be used to identify parallel relationships between activities [9], [10]. Aalst proposed α Algorithm by using Petri nets to represent process models [11], but is could not solve the loop. In Ref. [1,2], $\alpha+$ algorithm was proposed to solve the short loop of length 2 and the short loop of length 1 in the different stage of the algorithm. For invisible tasks, the $\alpha\#$ algorithm was proposed by Tsinghua University research team [4]. The $\alpha++$ algorithm [3] gives definitions of direct and indirect dependencies, using this new relationship to solve the structure of non-free choice. Guo et al. proposed$\alpha\$$ algorithm to mine hidden tasks in a non-free choice structure [12]. The β algorithm [13] has two new event types compared to the α algorithm: START and COMPLETE. We could use these two types of events to easily judge the concurrent structure by the temporal relationship. The lambda algorithm [14] proposes a new event log structure: multiple sets of events, and uses this kind of log structure to reduce data size and performance loss. The class of α algorithm is a good illustration of the key ideas behind process mining, but the α algorithm and its improved algorithm have their inherent disadvantages. In addition, David et al. proposed WoMine, which can retrieve frequent behavioral patterns such as sequences, selections, parallels and loops from the model [31]. Niek Tax et al. quantified over-approximation in a consistent way for all models and logs [29]. Dino Knoll et al. proposed domain ontology to support process mining within internal logistics [32]. First, they require that the event log must be complete. In addition, these algorithms are sensitive to uncertain data. The ability to deal with uncertain data is weak.

Most of the traditional process mining algorithms have the limitations that the log must be complete, and better analyzing the model with complex parallel structure is based on complete log data. How to mine incomplete logs, especially with uncertain data, is a difficult problem. Leemans *et al.* [15] analyzed the impact of incomplete logs on process mining and introduced a probabilistic behavioral relationship that is less sensitive to incompleteness, thus proposing a more robust algorithm. The technique proposed by Song *et al.* [16] uses the relationship of activity dependence to reduce the requirement for log completeness. In dealing with uncertain logs, Hwang *et al.* [17] proposed an algorithmic framework that adds noise-tolerant mechanisms to resist real-world noise. A causal network [5], [6] was used to describe the process model for heuristic process mining. The heuristic process mining considered the frequency of event trajectories in the construction process model. However, complex routing structures could not be described by the causal network, which has not sophisticated and mature analysis tools. Since the heuristic algorithm of process mining does not give a suitable threshold for removing uncertain data. The threshold set by user subjectively will not remove the uncertain data correctly, and will also cause the removal of the correct arc. In Ref.[18], the heuristic process mining algorithm was improved by computing the average dependence metric to confirm the threshold interval. Maruster *et al.* [19] used a logistic regression model to find a process model from an incomplete log with uncertain data, which can find 95% direct dependence in the presence of parallelism. Cheng and Kumar [20] proposed a method for constructing a classifier on the subset of logs and using a classification method to delete noise trajectories. Conforti *et al.* [21] proposed BPMN Miner to discover the hierarchical BPMN model. The method uses an approximation function and an inclusion-dependent discovery technique to identify the effects produced by uncertain data, and proved to withstand noise under different levels.

In recent years, some researchers have introduced the concept of entropy into the process mining and conformance checking. Bose et al. introduced the entropy formula to detect the process drift [23]. Bose et al. introduced entropy for trace alignment in the process diagnostics. Cook and Wolf [7] combined entropy with event types, periodicity, and causality to discover parallel behavior. But it can't find a parallel structure only by entropy, nor does the related literature provide a way to generate clear process models. And it can't solve the noise problem. In recent years, entropy is rarely used to process log noise in process mining. In this paper, the maximum entropy principle is applied to solve the problem of selecting the threshold for noise removal in process mining.

In conclusion, the existing process mining algorithms cannot deal with the problem caused by uncertain data well. In this paper, an improved correlation matrix and the maximum entropy principle are used to give a method to determine the threshold of processing uncertain data. It is used to remove the noise from the log. Finally, an algorithm framework for noise removal adaptively is given. Besides, according to the frequency of activity dependencies in the even log, we provide a method to identify the parallel/selective structure and can gain a process model based on Petri net through the directed graph.

III. PROCESSING UNCERTAIN LOG DATA

In this section, we first provide an activity sequence based on the event logs. Secondly, a counterexample is given based on the frequency matrix. It is noted that it is insufficient to use the frequency threshold alone to remove uncertain data. The method of determining the threshold of uncertain data is given by information entropy. Besides, the rule of determining parallel structure is given to remove the impact of parallel imbalances while calculating the threshold. The rule of determining the selective structure is given to reach a process model based on Petri nets from the directed graph.

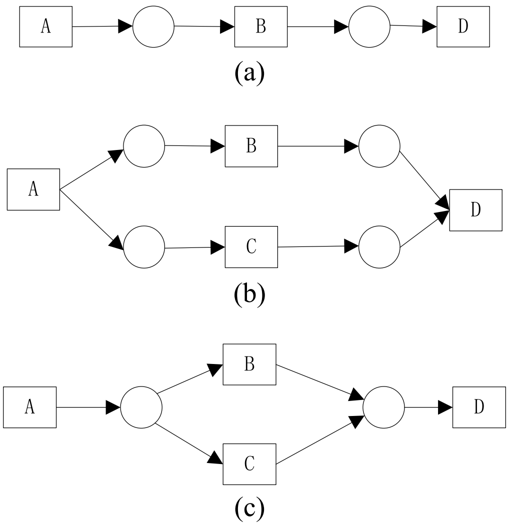


FIGURE 1. (a). Sequential structure. (b). Parallel structure. (c). Selective structure.

A. DEFINITIONS

Let L denote an event log and T denote the set of task.

Definition 1 (Direct Dependency):

Let $\sigma = t_1 t_2 \ldots t_m$ denote a trajectory with length m. A direct dependency will be gained from activity a to b if and only if $t_i = a, t_{i+1} = b, i \in \{1, 2, \ldots \mathrm{m} - 1\}$, represented as $a \to b$. a is a direct precursor of b, represented as $\cdot b = a$; and b is a direct successor of a, represented as $a\cdot = b$ [11].

Definition 2 (Loop):

A loop from activities t_i *and* t_j if and only if $t_i \to t_j$ and $t_j \to t_i$, represented as $t_i \leftrightarrows t_j$[11].

Definition 3 (Frequency of Direct Dependency):

Count$(t_i, t_j) = \sum_{\sigma \in \mathrm{L}} |\{1 \le i \le |\sigma| | \sigma(\mathrm{i}) = \mathrm{a} \bigwedge \sigma(\mathrm{i}+1) = \mathrm{b}\}|$ [31]. This represents the number of times the direct dependency between activity t_i *and* t_j is observed in the log.

For the workflow net, there are three basic structures sequential structure, parallel structure, and selective structure as shown in Fig.1.

B. FREQUENCY MATRIX AND FREQUENCY THRESHOLD

A frequency matrix is formed by the frequency of the direct dependency between all activities in the log.

Definition 4 (Frequency Matrix):

If there are n tasks in the log, we can form the frequency matrix of $n \times n$, represented as FM. Let FM_{ij} $(0 < i < n, 0 < j < n, i \in Z) =$ Count(t_i, t_j). This represents the number of direct dependency between activities t_i *and* t_j observed in the log.

For example, prototype the model is shown in Fig.2.

The model can generate data as follows: $\{< A, B, D, E, F >^{28}, < A, C, D, E, F >^{29}, < A, B, E, D, F >^{31}, < A, C, E, D, F >^{30}, < A, D, B, E, F >^{31}, < A, D, C, E, F >^{29}\}$. Among these, $< A, D, C, E, F >^{29}$ represents that the trace $< A, D, C, E, F >$ appears 29 times. According to definition

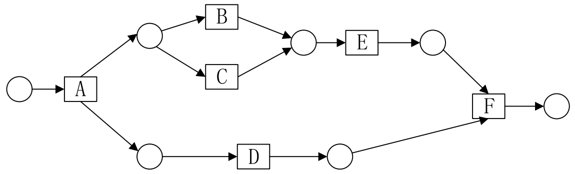


FIGURE 2. **Prototype Model.**

4, the frequency matrix FM is formed as follows:

$$\left(\begin{array}{ccccccc} & A & B & C & D & E & F \\ A & 0 & 59 & 59 & 60 & 0 & 0 \\ B & 0 & 0 & 0 & 28 & 62 & 0 \\ C & 0 & 0 & 0 & 29 & 59 & 0 \\ D & 0 & 31 & 29 & 0 & 57 & 61 \\ E & 0 & 0 & 0 & 30 & 0 & 117 \\ F & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

We detect the parallel, sequential, and selective structure according to the frequency matrix. However, when there is uncertain data in the log, our recognition of the model structure is hindered. Uncertain data refers to the rare and infrequent trace recorded in the log. For instance, if there is a trace $< A, B, C, E, F >^2$, the error recorded in the above log forms the uncertain data: $\{< A, B, D, E, F >^{28}$, $< A, C, D, E, F >^{29}$, $< A, B, E, D, F >^{31}$, $< A, C, E, D, F >^{30}$, $< A, D, B, E, F >^{31}$, $< A, D, C, E, F >^{29}$, $< A, B, C, F >^2$, $< A, E >^6\}$. If the uncertain data is not processed, the frequency matrix FM is as follows:

$$\left(\begin{array}{ccccccc} & A & B & C & D & E & F \\ A & 0 & 61 & 59 & 60 & 6 & 0 \\ B & 0 & 0 & 2 & 28 & 62 & 0 \\ C & 0 & 0 & 0 & 29 & 59 & 2 \\ D & 0 & 31 & 29 & 0 & 57 & 61 \\ E & 0 & 0 & 0 & 30 & 0 & 117 \\ F & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

There is a mistaken recognition of the direct dependency between activities B and C. To remove uncertain data, the influence degree in Ref.[9] is used as a threshold. However, there is a shortcoming in threshold selection. We provide its definition and give an example to assess its unreasonableness.

Definition 5 (Frequency Threshold):

The definition of the frequency threshold in the literature [9] is given as follows.

$$\delta_{Frequence} = 1 + \mathrm{Round}(\mathrm{N}*\frac{m}{n}) \tag{1}$$

where *Round* indicates the rounding of a number; N indicates the uncertain data factor with a default value of 0.05; m is the number of different tasks, and n is the number of trace lines in the log.

In [9], the frequency between activities is compared with the above threshold and the dependency less than the threshold will be removed as uncertain data. This method is acceptable but lacks adaptability in threshold selection. Because the occurrence of uncertain data cannot be predicted, the imbalance of the uncertain data itself will affect the judgment of normal dependency. In the above example, for the above example, the frequency threshold of the noisy log can be calculated as $\delta_{Frequence} = 1 + Round\left(0.05 * \frac{186}{6}\right) = 3.\mathrm{Count}\,(\mathrm{B}, \mathrm{C}) = 2 \leq 3, \mathrm{Count}\,(\mathrm{C}, \mathrm{F}) = 2 \leq 3$. The direct dependency between B and C, C and F are caused by uncertain data should be removed. However, there is still a dependency caused by uncertain data in FM: A → E. Therefore, it is insufficient to simply remove uncertain data by the frequency threshold. We introduce a correlation measure and integrate information entropy to determine the correlation threshold.

C. THE IMPROVED FREQUENCY MATRIX AND CORRELATION MATRIX

Due to the uncertain data in logs, the analysis of the frequency matrix in the Ref.[25] will lead to erroneous judgment of uncertain data. We need to improve the original frequency matrix. In addition, before determining the correlation threshold, we need to prejudge the parallel structure in advance to remove the influence of the imbalance between parallel activities while calculating the threshold. Therefore, the judgment of the parallel relation should also be reflected in the improved frequency matrix.

To prevent activity dependency, which is greater than the frequency threshold in uncertain data affecting the judgment of parallel structure, we must define the decision threshold for parallel structures. This is the average value of the none-zero elements in the frequency matrix FM.

Definition 6 (Parallel Structure Decision Threshold):

Parallel structure decision threshold is defined as $\bar{f}$ shown in equation 2.

$$\bar{f} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} FM_{ij}}{\Delta\,(FM)} \tag{2}$$

where $\Delta\,(FM)$ is the number of elements in the matrix FM that are not 0; that is, the number of pairs of activities with a directly dependency in the log. $FM_{ij} = \mathrm{Count}(t_i, t_j)$. It is the number of direct dependency between activities t_i *and* t_j.

Rule 1 (Decision Rules of Parallel Structure):

For frequency matrix FM, $\forall \mathrm{i}, \mathrm{j} \in [1, \mathrm{n}]\,, \mathrm{i} \neq \mathrm{j}$, if there is a parallel structure between activities $a_i andb_j$ if and only if the two elements $FM_{ij} > \delta_{Frequence}$ and $FM_{\mathrm{ji}} > \delta_{Frequence}$ and $FM_{ij}{+}FM_{\mathrm{ji}} > \bar{f}$ and $FM_{ij}{*}FM_{ji} \neq 0$, denoted as $a_i||a_j$.

Definition 7 (Improved Frequency Matrix):

For frequency matrix FM and its elements FM_{ij}, let $FM_{ij} = 0$ if and only if $FM_{ij} < \delta_{Frequence}$ or $t_i||t_j$. The improved frequency matrix $FM^{'}$ can be obtained.

If the frequency directly dependent on the activities is less than the frequency threshold, the corresponding element in the frequency matrix is set to 0. Similarly, if there is a parallel structure among activities, the corresponding element in the frequency matrix is set to 0. Then the proposed frequency matrix is $FM^{'}$.

In the above example, $FM_{\mathrm{BD}} = 28, FM_{DB} = 31, FM_{\mathrm{CD}} = 29, FM_{\mathrm{DC}} = 29, FM_{\mathrm{DE}} = 57, FM_{\mathrm{ED}} = 30, \bar{f} = \frac{695}{16} = 43.4$. There is $FM_{\mathrm{BD}}, FM_{\mathrm{DB}} > \delta_{Frequence}$ and $FM_{\mathrm{BD}} * FM_{\mathrm{DB}} \neq 0; FM_{\mathrm{CD}}, FM_{\mathrm{DC}} > \delta_{Frequence} and FM_{\mathrm{CD}} * FM_{\mathrm{DC}} \neq 0; FM_{\mathrm{DE}}, FM_{\mathrm{ED}} > \delta_{Frequence}$ and $FM_{\mathrm{DE}} * FM_{\mathrm{ED}} \neq 0$. There is B| |D, C| $|D, D||E$. The corresponding element is set to 0, and the proposed frequency matrix $FM^{'}$ is as follows.

$$\left(\begin{array}{ccccccc}
 & A & B & C & D & E & F \\
A & 0 & 61 & 59 & 60 & 6 & 0 \\
B & 0 & 0 & 2 & 0 & 62 & 0 \\
C & 0 & 0 & 0 & 0 & 59 & 2 \\
D & 0 & 0 & 0 & 0 & 0 & 61 \\
E & 0 & 0 & 0 & 0 & 0 & 117 \\
F & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right)$$

The uncertain data cannot be completely removed according to the frequency threshold. After obtaining parallel activities, we need to calculate the correlation between activities based on the improved frequency matrix to construct the correlation matrix. Then we can use the maximum entropy principle to determine the threshold for processing uncertain log data according to the correlation matrix.

Definition 8 (Correlation Measure Based on the Improved Frequency Matrix):

In this study, we provide an improved correlation measure formula based on the improved frequency matrix measure.

$$CM_{ij} = \frac{FM^{'}_{ij} - FM^{'}_{ji}}{FM^{'}_{ij} + FM^{'}_{ji} + 1} \tag{3}$$

The correlation measure is regarded as the element value of the correlation matrix, and the correlation matrix CM is constructed as follows.

$$\left(\begin{array}{ccccccc}
 & A & B & C & D & E & F \\
A & 0 & 0.9839 & 0.9833 & 0.9836 & 0.8571 & 0 \\
B & 0 & 0 & 0.6667 & 0 & 0.9841 & 0 \\
C & 0 & 0 & 0 & 0 & 0.9833 & 0.6667 \\
D & 0 & 0 & 0 & 0 & 0 & 0.9839 \\
E & 0 & 0 & 0 & 0 & 0 & 0.9917 \\
F & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right)$$

The correlation measure will be used to calculate the information entropy.

D. DETERMINE THRESHOLD FOR PROCESSING UNCERTAIN LOG DATA BY MAXIMUM ENTROPY PRINCIPLE

The existing process mining algorithm cannot effectively solve the problem of uncertain log data; there is a shortcoming as previously shown. The existing algorithms intended to solve the uncertain data problem often have strong pertinence. Further, the fixed threshold method is overly dependent on the parameter setting, and the universality is generally low.

In the field of image processing, threshold segmentation is common in image segmentation. By selecting the appropriate threshold, the target and background in the original image are separated, which provides a basis for subsequent classification and recognition. The key to threshold segmentation is threshold selection. Because entropy represents the average information, some scholars use the concept of entropy to select the segmentation threshold [26]. In image processing, the aim of this method is to divide the gray histogram of the image into different classes so that the total entropy of all kinds is the maximum. From the perspective of the information theory, this approach obtains the maximum amount of information. The threshold segmentation method in image segmentation is introduced into process mining in this study. The traces in the log are composed of normal traces and noisy traces, and the multiple thresholds in the image processing have degenerated to a single threshold. The threshold for processing uncertain log data is decided based on the principle of maximum entropy, which reveals the maximum amount of information contained in the log. This algorithm does not depend on the artificial setting and has strong adaptability.

The threshold definition based on the maximum entropy is reasonable. If the correlation value between activities fluctuates greatly, the corresponding entropy is small. The appropriate threshold is selected to divide the dependency between activities into two parts: a normal part and a part caused by uncertain data. The correlation values in the two parts are gentle, and the corresponding entropy is also large. On the contrary, if the threshold selection is inaccurate, some dependency will be classified into the wrong cluster, and the calculated entropy will be smaller than the entropy under the appropriate threshold value. Based on this principle, the optimal threshold is obtained, which maximizes the sum of the entropy of two parts.

Let $c_1, c_2, \ldots, c_n$ be the differentcorrelation values in CM. There are a total of n correlation values in total, arranged in ascending order. The probability distribution of the i-th correlation value is $p_{\mathrm{c}_i} = \frac{\mathrm{c}_i}{N}.\mathrm{N} = \sum_{i=1}^{n} |\mathrm{c}_i|$, representing the total number of values in CM.

If the threshold is set as the s-th CM value, two probability distributions will be obtained. A probability distribution contains the CM value $\mathrm{c}_1 \sim \mathrm{c}_{s-1}$ and is caused by uncertain data. Another probability distribution contains the CM value $\mathrm{c}_s \sim \mathrm{c}_n$ and is caused by normal traces. Two probability distributions are as follows.

$$\mathrm{A} : \frac{p_{\mathrm{c}_1}}{P_{\mathrm{c}_{s-1}}}, \frac{p_{\mathrm{c}_2}}{P_{\mathrm{c}_{s-1}}}, \ldots, \frac{p_{\mathrm{c}_{s-1}}}{P_{\mathrm{c}_{s-1}}}$$

$$\mathrm{B} : \frac{p_{\mathrm{c}_s}}{1 - P_{\mathrm{c}_{s-1}}}, \frac{p_{\mathrm{c}_{s+1}}}{1 - P_{\mathrm{c}_{s-1}}}, \ldots, \frac{p_{\mathrm{c}_n}}{1 - P_{\mathrm{c}_{s-1}}}$$

Among them, $P_{\mathrm{c}_{s-1}} = \sum_{i=1}^{s-1} p_{\mathrm{c}_i}, 1 - P_{\mathrm{c}_{s-1}} = \sum_{i=s}^{n} p_{\mathrm{c}_i}$.

According to the formula of information entropy, $\mathrm{H}_n = -\sum_{i=1}^{n} p_{\mathrm{c}_i} \ln p_{\mathrm{c}_i}$, and the respective entropies of the above two distributions are as follows.

$$
\begin{aligned}
\mathrm{H}\,(\mathrm{A}) &= -\sum_{i=1}^{s-1} \frac{p_{c_i}}{P_{c_{s-1}}} \ln \frac{p_{c_i}}{P_{c_{s-1}}} \\
&= -\frac{1}{P_{c_{s-1}}} \sum_{i=1}^{s-1} p_{c_i} (\ln p_{c_i} - \ln P_{c_{s-1}}) \\
&= -\frac{1}{P_{c_{s-1}}} \left(-H_{s-1} - P_{c_{s-1}} \ln P_{c_{s-1}}\right) \\
&= \ln P_{c_{s-1}} + \frac{H_{s-1}}{P_{c_{s-1}}}.
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{H}\,(\mathrm{B}) &= -\sum_{i=s}^{n} \frac{p_{c_i}}{1-P_{c_{s-1}}} \ln \frac{p_{c_i}}{1-P_{c_{s-1}}} \\
&= -\frac{1}{1-P_{c_{s-1}}} \sum_{i=s}^{n} p_{c_i} [\ln p_{c_i} - \ln\left(1-P_{c_{s-1}}\right)] \\
&= -\frac{1}{1-P_{c_{s-1}}} [H_{s-1} - H_n - (1-P_{c_{s-1}}) \ln (1-P_{c_{s-1}})] \\
&= \ln (1-P_{c_{s-1}}) + \frac{H_n - H_{s-1}}{1-P_{c_{s-1}}}
\end{aligned}
$$

According to the maximum entropy principle, let $\mathrm{h}\,(\mathrm{s}) = \mathrm{H}\,(\mathrm{A}) + \mathrm{H}\,(\mathrm{B}) = \ln P_{c_{s-1}} + \ln (1-P_{c_{s-1}}) + \frac{H_{s-1}}{P_{c_{s-1}}} + \frac{H_n - H_{s-1}}{1-P_{c_{s-1}}}$. The s value that maximizes the entropy is calculated as $\mathrm{s} = \mathrm{argmax}(\mathrm{h}\,(\mathrm{s}))$. Based on the principle of the maximum entropy, the threshold for uncertain data is set to c_s.There is no place that exists between activities in which the dependency is less than c_s; the dependency is caused by uncertain data.

According to the maximum entropy principle, the algorithm for determining the threshold for uncertain data is adaptive to logs; that is, no manual setting of parameters is needed, and different thresholds are obtained for different logs. The algorithm for determining the threshold for uncertain data is shown in Algorithm 1.

Algorithm 1 Threshold Algorithm for Uncertain Data

```
Input: event log $L$
Output: correlation threshold $\delta_{\mathrm{Correlation}}$
According to the definition in section 3.1, build the frequency matrix $FM$ of size $|\mathrm{L}| \times |\mathrm{L}|$.
Calculating the frequency threshold $\delta_{Frequence}$ based on definition 4.
According to rule 1, the parallel structure is judged, and the frequency matrix is improved to get $FM^{'}$.
Get the correlation matrix $CM$ according to definition 6,
The values of the $CM$ are arranged from small to large $c_1 \ldots c_n$
for($\mathrm{s} = 1; \mathrm{s} \leq \mathrm{n}; \mathrm{s}++$)
    Calculate the $h(s)$
Calculate $\mathrm{s} = \mathrm{argmax}(\mathrm{h}\,(\mathrm{s}))$
return $\delta_{\mathrm{Correlation}} = c_s$
```

The input of the above threshold algorithm for uncertain data is log L, and the correlation threshold of the log is returned. $|L|$ denotes the number of activities in the log.

TABLE 1. Correlation measure and its distribution.

i	c_i	p_{c_i}
1	0.6667	2/10
2	0.8571	1/10
3	0.9833	2/10
4	0.9836	1/10
5	0.9839	2/10
6	0.9841	1/10
7	0.9917	1/10

For the example presented in section 3.2, the values in the matrix CM are arranged in ascending order shown in Tab. 1.

According to the algorithm proposed in this section, the s value that maximizes the entropy is 3; that is, $\mathrm{s} = \mathrm{argmax}\,(\mathrm{h}\,(\mathrm{s})) = 3$. Then the threshold is $\delta_{\mathrm{Correlation}} = c_3 = 0.9833$.

IV. AN ADAPTIVE DEALING WITH UNCERTAIN DATA ALGORITHM FRAME BASED ON INFORMATION ENTROPY

In this section, we give a detailed description of the adaptive dealing with uncertain data algorithm based on information entropy. We provided the decision rules for parallel structures in section 3, along with the rule of judging the selective structure and loops.

Rule 2 (The Determining Rule of Selective Structure):

For activities $t_k, t_i, t_j \in T$, k ,i, j ∈ [1, n], there is a selective structure between t_i and t_j if and only if $FM^{'}_{ki} > \delta_{Frequence}$ and $CM_{ki} > \delta_{\mathrm{Correlation}}$; $FM^{'}_{kj} > \delta_{Frequence}$ and $CM_{kj} > \delta_{\mathrm{Correlation}}$. There is no parallel structure between t_i and t_j, denoted as $t_i \odot t_j$。

Rule 3 (The Determining Rule of Loops):

For the improved frequency matrix $FM^{'}$, ∀i ∈ [1, n], there is a loop between t_i and t_i if and only if $FM_{ii} > \delta_{Frequence}$, denoted as $t_i \circlearrowleft t_i$.

Next, the definition of the beginning and ending task in the log is given.

Definition 9 (First, Last [30]):

Let L denote the event log for task set $T.\sigma = t_1 t_2 \ldots t_n$ is a trace with length n, First $(\sigma) = t_1$.Last $(\sigma) = t_n$, referring to the beginning and end of a trace.

Next, we provide a description of the algorithm in detail. The event log is the input, and a process model based on Petri nets will be gained. The detailed steps are shown in Algorithm 2.

This algorithm constructs the workflow based on Petri nets. Tasks will be extracted from the event logs in Step 1. These activities will eventually become transitions to the workflow net, and the set of beginning tasks and ending tasks will be extracted in Step 2 and step 3. T_I is the set of beginning tasks (i.e., the collection of all activities that occurs in the first trajectory position). T_O is the set of ending tasks (i.e., the collection of all activities that occurs in the last trajectory position). The frequency matrix FM will be reached according to the definition in section 3.1. Step 5 calculates the

Algorithm 2 An Adaptive Dealing With Uncertain Data Algorithm Based on Information Entropy

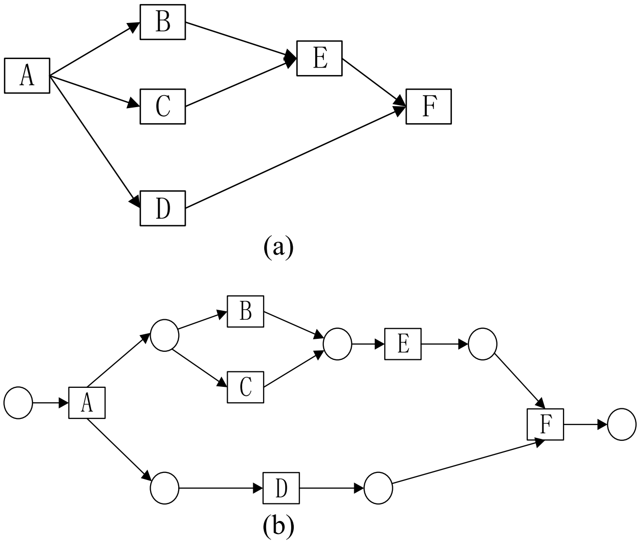
```
Input: event log $L$, composed of traces $\sigma \in L$
Output: workflow net
1. $T_L = \{t \in T | \exists_{\sigma \in L} t \in \sigma$
2. $T_I = \{t \in T | \exists_{\sigma \in L} t \in First(\sigma)$
3. $T_O = \{t \in T | \exists_{\sigma \in L} t \in Last(\sigma)$
4. Calculating F $M$
5. Calculating the frequency threshold $\delta_{Frequence}$ based on definition 6
6. Judging parallel structures based on rule 1
7. Calculating $FM^{'}$ and $CM$. Calculating the correlation threshold $\delta_{\text{Correlation}}$
8. Judging selective structures based on rule 3
9. For activities $t_i, t_j \in T_L$, if corresponding $FM^{'}_{ij} > \delta_{Frequence}$ and $CM_{ij} > \delta_{\text{Correlation}}$,
    then there is a directed arc from $t_i$to$t_j$. Eventually form a directed graph $DG$.
10. For activities $t_i, t_j \in T_L$, if $t_i \odot t_j$, then there is a selective structure between $t_i$ and $t_j$.
      Refactoring them into a selective fragment $Fragment_{ij}$. The arcs before and after thefragment is merged to form a digraph $DG^{'}$.
11. For each arc of $DG^{'}$, add a place in the middle of it.
12. For the selective fragment $Fragment_{ij}$ in step 10, restore it to activities $t_i, t_j$. Add arcs between them and places.
13. Add the unique source library $i_L$ and link to the activities in $T_I$. Add the unique sink place $o_L$ and link to the activities in $T_O$.
14. Return the workflow net
```

frequency threshold $\delta_{Frequence}$ based on definition 4. In step 6, the parallel structure is judged according to rule 1. The value of FM is compared with the frequency threshold $\delta_{Frequence}$. $FM^{'}$ and CM is calculated in step 7. Then the correlation threshold $\delta_{\text{Correlation}}$ will be calculated. In step 6, the selective structure is judged according to rule 3.Step 9constructs the directed graph DG based on the above two thresholds for dealing with uncertain data. Step 10 reconstructs the selective structure into a fragment and adjusts the directed graph to DG$^{'}$. Step 11 to step 13 convert the directed graph into a Petri net.

For the example given in section 3, we can obtain the directed graph as shown in Fig.3 (a) based on the above algorithm. Because there is a selective structure between B and C, a process model based on a Petri net transformed from the directed graph is shown in Fig.3(b).

V. EXPERIMENT

This section analyzes the effect of fitness for dealing with uncertain data, appropriateness and the different uncertain data ratios. The proposed algorithm is compared with the $\alpha + +$ algorithm, heuristic algorithm and genetic process mining to verify its effectiveness.



(a)

(b)

FIGURE 3. (a). The directed graph obtained from the algorithm. (b). Petri net transformed from a directed graph.

A. EVALUATION CRITERIA

This paper uses fitness and behavioral appropriateness as the evaluation criteria of the process model.

In this study, fitness is calculated by token play. The range of fitness is [0,1]. The higher the fitness, the higher the log traces and model match. Rozinat and Van der Aalst [27] gave the fitness algorithm based on token play. k is the number of different trajectories in the log. For every trajectory i (1 $\leq$ i $\leq$ k), n_i is the number of the current trajectory appears in the log, m_i is the number of missing tokensduringthe token play of the current trajectory, r_i is the number of remaining tokens during the token play of the current trace, c_i is the number of consumed tokensduringthe token play of the current trace,and p_i is the number of emerged tokens during the token play of the current trajectory. The definition of *fitness* based on token is as follows.

$$\text{Fitness} = \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i}\right) + \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i}\right) \quad (4)$$

Behavioral appropriateness reflects how accurately the model describes the log traces. It requires that the model's behavior is as close as possible to that in the log [3].In this study, we use the advanced behavioral appropriateness in [28] to calculate the quality of each algorithm's mining results. It defines two relationships: follows relations and precedes relations.

1) FOLLOWS RELATIONS

Two activities (x, y) are in "Always Follows", "Never Follows", or "Sometimes Follows" relation in the case that, if x is carried out at least once, then always, never, or sometimes y is finally carried out, respectively. They are represented by A_F, N_F, S_F respectively.

TABLE 2. Activity number.

Activity	Number
NEW	node0
CODE OK	node1
STORNO	node2
REOPEN	node3
REJECT	node4
CHANGE END	node5
FIN	node6
RELEASE	node7
BILLED	node8
DELETE	node9
ZDBC_BEHAN	node10
JOIN-PAT	node11
CODE NOK	node12
CODE ERROR	node13
MANUAL	node14
SET STATUS	node15
EMPTY	node16
CHANGE DIAGN	node17

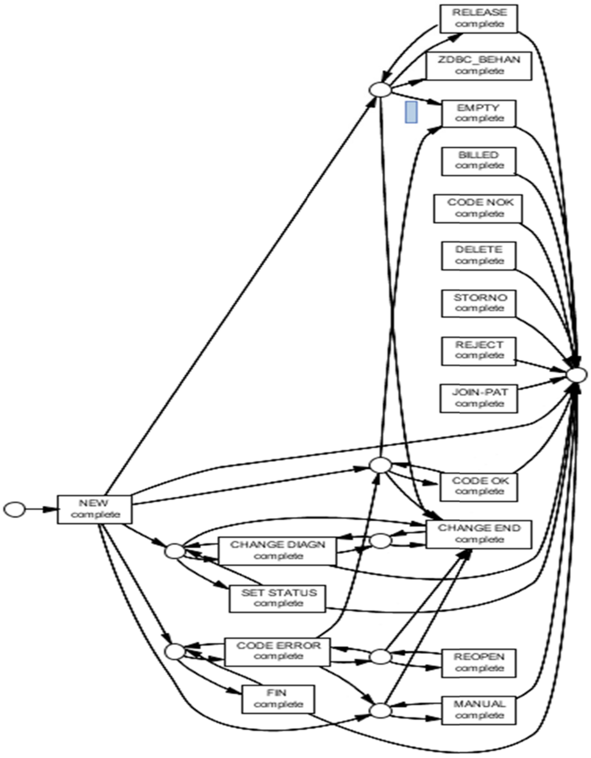


FIGURE 4. The process model discovered by $\alpha++$ algorithm.

2) PRECEDES RELATIONS

Two activities (x, y) are in “Always Precedes”, “Never Precedes”, or “Sometimes Precedes” relation in the case that, if y is carried out at least once, then always, never, or sometimes x was carried out sometime before respectively. They are represented by A_P, N_P, S_P respectively.

TABLE 3. Correlation measure distribution.

i	c_i	p_{c_i}
1	0.99726776	1/20
2	0.99728261	1/20
3	0.99769053	1/20
4	0. 99777283	1/20
5	0. 99780702	1/20
6	0. 99833333	1/20
7	0. 99876084	1/20
8	0.9988024	1/20
9	0.99910634	1/20
10	0.99919734	1/20
11	0.99939173	1/20
12	0.999501	1/20
13	0.99957983	1/20
14	0.99968808	1/20
15	0.99979437	1/20
16	0.99996767	1/20
17	0.99997549	1/20
18	0.99997635	1/20
19	0.99998491	1/20
20	0.99998577	1/20

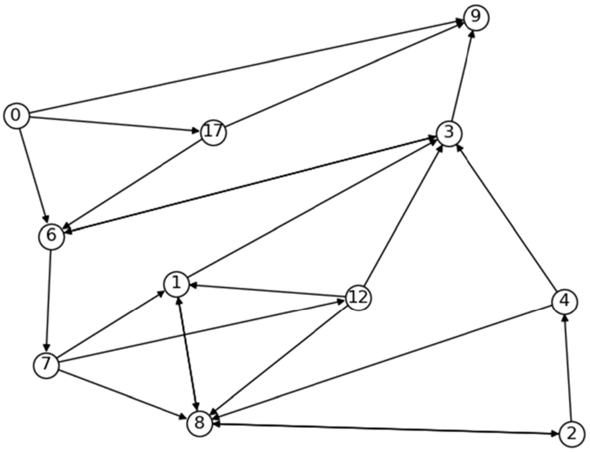


FIGURE 5. The directed graph.

Let S_F^M be the “Sometimes Follows” relation and S_P^M be the “Sometimes Precedes” relation in the process model, and S_F^L be the “Sometimes Follows” relation and S_P^L be the “Sometimes Precedes” relation in the event log. The advanced behavioral suitability metric a_B is defined as follows:

$$a_B = \frac{|S_F^M \cap S_F^L|}{2 * |S_F^M|} + \frac{|S_P^M \cap S_P^L|}{2 * |S_P^M|} \tag{5}$$

The range of a_B is (0, 1]. The greater the value is, the closer the model’s behavior to the log.

B. EXPERIMENTS ON HOSPITAL BILLS

The open source data set obtained from Ref.[28], was published on August 1, 2017, and provided by the Eindhoven

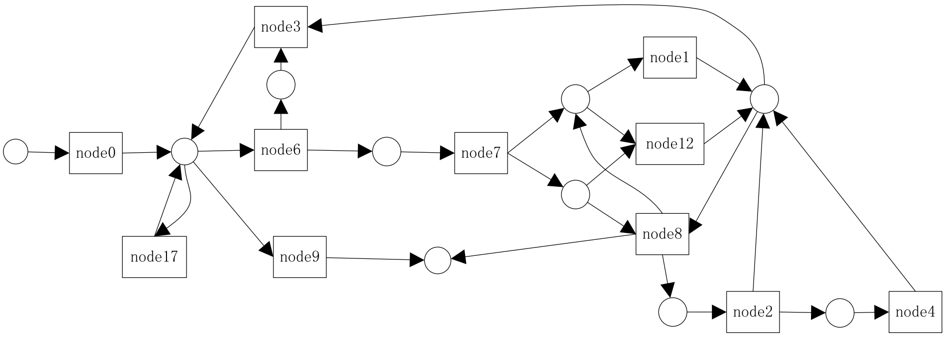


FIGURE 6. The process model.

TABLE 4. Different uncertain data ratio.

Log	L1	L2	L3	L4	L5	L6	L7	L8	L9
uncertain data ratio	15.9%	18.8%	20%	14.7%	9.3%	22.2%	15.9%	15.9%	24.7%

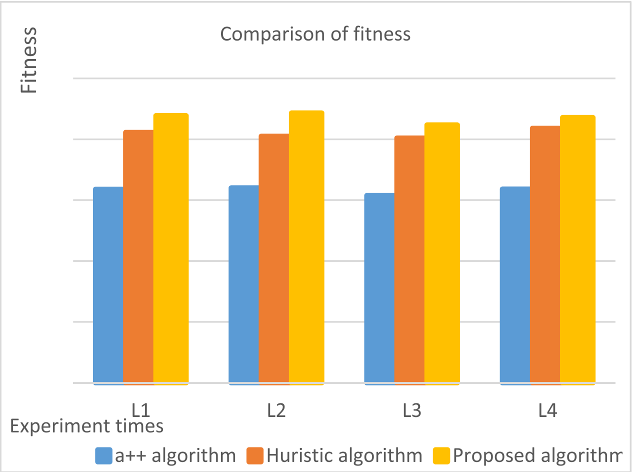


FIGURE 7. Comparison of fitness.

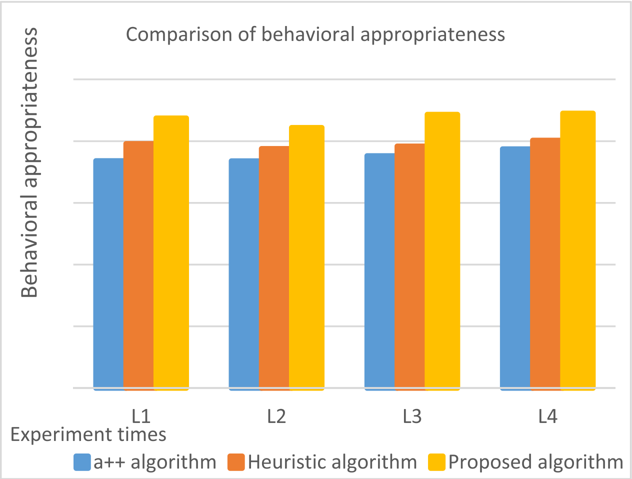


FIGURE 8. Comparison of behavioral appropriateness.

University of technology. We used the event log of 'Hospital Billing', which gained from the financial modules in ERP system of a regional hospital. The event log is primarily about events related to the billing for medical services. Each trace of the event log contains execution activity of the billing-related medical service, but the information about actual medical services is not presented in it. The 101289 traces in the event log are a random sample of process instances t recorded over three years.

The events are numbered, as shown in Table 2.

The process model gained by $\alpha ++$ algorithm was shown in Fig.4.

The fitting of the model is 0.667.

We mine the event log by our algorithm. The value of the correlation measure is arranged as shown in Table 3.

We can get the threshold $c_4 = 0.99777283$. According to the algorithm, the directed graph was obtained as shown in Fig.5 and the process model as shown in Fig.6.

The fitting of the model is 0.882. More experiments are carried out. The comparison of fitness is shown in Fig.7 and the comparison of behavioral appropriateness is shown in Fig.8. Our proposed algorithm has higher fitness and behavioral appropriateness.

C. ANALYSIS OF THE PROCESSING EFFECT OF DIFFERENT UNCERTAIN DATA RATIOS

In this paper, logs with different uncertain data ratios are selected for analysis. Let $|\sigma_{nomal}|$ be the number of normal traces in the log and $|\sigma_{noise}|$ be the number of noisy traces in the log. The uncertain data ratio of the log is: $\frac{|\sigma_{noise}|}{|\sigma_{nomal}|+|\sigma_{noise}|}$. The uncertain data ratio of the log in this paper is shown in Table 4.

For the log with the same uncertain data ratio, the average fitness value is recorded; the result appears in Fig.18.

As shown in Fig.9, our proposed algorithm has a good effect in mining the logs of different uncertain data ratios.

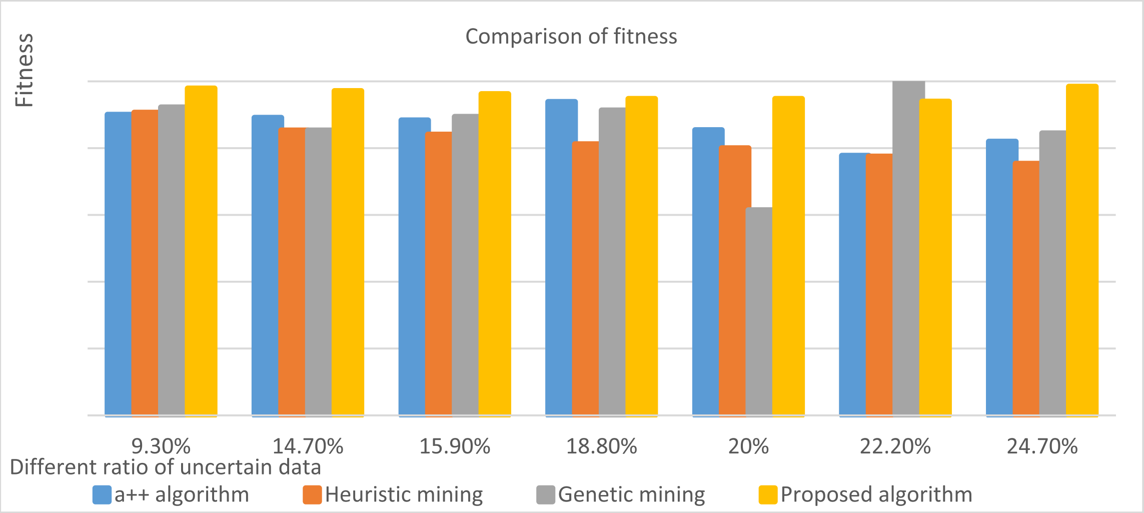


FIGURE 9. Comparison of fitness on the different ratio of uncertain data.

When the uncertain data ratio is 22.20%, the proposed algorithm is slightly worse than genetic process mining. However, genetic process mining is time-consuming. For the rest of the cases, the proposed algorithm has better fitness than the other tested algorithms.

VI. CONCLUSION

In this paper, a process mining algorithm for adaptive uncertain data removal is proposed. The proposed method first extracts the frequency of the direct dependence of the activity from the event log and constructs an improved frequency matrix. We note that it is insufficient to only use the frequency threshold to solve the problem of uncertain data. We improve the correlation measure in heuristic mining to build the correlation matrix based on the improved frequency matrix. Combined with the most commonly used maximum entropy principle in the image domain, a self-adaptive method is proposed to determine the threshold to remove the uncertain data relationship in logs. In this paper, the parallel/selective structure is identified by the threshold for processing uncertain data, and an algorithm framework or adaptive uncertain data removal is given. The algorithm proposed in this paper is independent of artificial parameter-setting and is highly adaptable. The process mining algorithm based on information entropy represents a new attempt to remove uncertain data in process mining. Experiments show that the proposed algorithm has better fitness, behavioral appropriateness, and structural appropriateness when mining logs with uncertain data.

REFERENCES

[1] A. K. A. de Medeiros, B. F. van Dongen, W. M. P. van der Aalst, and A. J. M. M. Weijters, ‘‘Process mining: Extending the alpha-algorithm to mine short loops,’’ Techn. Univ. Eindhoven, Eindhoven, The Netherlands, BETA Working Papers, 2004, vol. 113.

[2] A. K. A. de Medeiros, W. M. P. van der Aalst, and A. J. M. M. Weijters, ‘‘Workflow mining: Current status and future directions,’’ in *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. OTM* (Lecture Notes in Computer Science), vol. 2888, R. Meersman, Z. Tari, and D. C. Schmidt, Eds. Berlin, Germany: Springer, 2003.

[3] L. Wen, W. van der Aalst, J. Wang, and J. Sun, ‘‘Mining process models with non-free-choice constructs,’’ *Data Mining Knowl. Discovery*, vol. 15, no. 2, pp. 145–180, 2007.

[4] L. Wen, J. Wang, W. M. P. van der Aalst, B. Huang, and J. Sun, ‘‘Mining process models with prime invisible tasks,’’ *Data Knowl. Eng.*, vol. 69, no. 10, pp. 999–1021, 2010.

[5] A. J. M. M. Weijters, W. M. P. van der Aalst, and A. K. A. de Medeiros, ‘‘Process mining with the HeuristicsMiner algorithm,’’ Techn. Univ. Eindhoven, Eindhoven, The Netherlands, BETA Working Papers, 2006, vol. 166.

[6] A. J. M. M. Weijters and J. T. S. Ribeiro, ‘‘Flexible heuristics miner (FHM),’’ in *Proc. IEEE Symp. Comput. Intell. Data Mining (CIDM)*, Apr. 2011, pp. 310–317.

[7] J. E. Cook and A. L. Wolf, ‘‘Automating process discovery through event-data analysis,’’ in *Proc. 17th Int. Conf. Softw. Eng.*, Seattle, Washington, USA, Apr. 1995, p. 73.

[8] R. Agrawal, D. Gunopulos, and F. Leymann, ‘‘Mining process models from workflow logs,’’ in *Proc. Int. Conf. Extending Database Technol.*, Valencia, Spain, 1998, pp. 467–483.

[9] S. S. Pinter and M. Golani, ‘‘Discovering workflow models from activities’ lifespans,’’ *Comput. Ind.*, vol. 53, no. 3, pp. 283–296, 2004.

[10] M. Golani and S. S. Pinter, ‘‘Generating a process model from a process audit log,’’ in *Proc. Int. Conf. Bus. Process Manage.* Springer, Berlin, Germany, 2003, pp. 136–151.

[11] W. van der Aalst, T. Weijters, and L. Maruster, ‘‘Workflow mining: Discovering process models from event logs,’’ *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 9, pp. 1128–1142, Sep. 2004.

[12] Q. Guo, L. Wen, J. Wang, Z. Yan, and P. S. Yu ‘‘Mining invisible tasks in non-free-choice constructs,’’ in *Proc. Int. Conf. Bus. Process Manage.* Springer, Cham, Switzerland, 2016, pp. 109–125.

[13] L. Wen, J. Wang, W. M. P. van der Aalst, B. Huang, and J. Sun, ‘‘A novel approach for process mining based on event types,’’ *J. Intell. Inf. Syst.*, vol. 32, no. 2, pp. 163–190, 2009.

[14] D. Wang, J. Ge, H. Hu, B. Luo, and L. Huang, ‘‘Discovering process models from event multiset,’’ *Expert Syst. With Appl.*, vol. 39, no. 15, pp. 11970–11978, 2012.

[15] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, ‘‘Discovering block-structured process models from incomplete event logs,’’ in *Proc. Int. Conf. Appl. Theory Petri Nets Concurrency*. Springer, Cham, Switzerland, 2014, pp. 91–110.

[16] W. Song, H.-A. Jacobsen, C. Ye, and X. Ma, ‘‘Process discovery from dependence-complete event logs,’’ *IEEE Trans. Serv. Comput.*, vol. 9, no. 5, pp. 714–727, Sep./Oct. 2016.

[17] S.-Y. Hwang and W.-S. Yang, ‘‘On the discovery of process models from their instances,’’ *Decis. Support Syst.*, vol. 34, no. 1, pp. 41–57, Dec. 2002.

[18] R. Sarnom, F. Haryadita, D. Sunaryono, and A. Munif, ‘‘Model discovery of parallel business processes using modified heuristic miner,’’ in *Proc. Int. Conf. Sci. Inf. Technol. (ICSITech)*, Oct. 2015, pp. 30–35.

[19] L. Maruster, A. J. M. M. T. Weijters, W. M. P. W. van der Aalst, and A. van den Bosch, "Process mining: Discovering direct successors in process logs," in *Proc. Int. Conf. Discovery Sci.* Springer, Berlin, Germany, 2002, pp. 364–373.

[20] H.-J. Cheng and A. Kumar, "Process mining on noisy logs—Can log sanitization help to improve performance?" *Decis. Support Syst.*, vol. 79, pp. 138–149, Nov. 2015.

[21] R. Conforti, M. Dumas, L. García-Bañuelos, and M. L. Rosa, "BPMN Miner: Automated discovery of BPMN process models with hierarchical structure," *Inf. Syst.*, vol. 56, pp. 284–303, Mar. 2016.

[22] R. P. J. C. Bose, W. M. P. van der Aalst, I. Žliobaitė, and M. Pechenizkiy, "Handling concept drift in process mining," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.* Springer, Berlin, Germany, 2011, pp. 391–405.

[23] R. P. J. C. Bose and W. van der Aalst, "Trace alignment in process mining: Opportunities for process diagnostics," in *Business Process Management. BPM* (Lecture Notes in Computer Science), vol. 6336, R. Hull, J. Mendling, and S. Tai, Eds. Berlin, Germany: Springer, 2010.

[24] J. E. Cook and A. L. Wolf, "Event-based detection of concurrency," *ACM SIGSOFT Softw. Eng. Notes*, vol. 23, no. 6, pp. 35–45, 1998.

[25] A. J. M. M. Weijters and W. M. P. van der Aalst, "Process mining: Discovering workflow models from event-based data," in *Proc. 13th Belgium-Dutch Conf. Artif. Intell. (BNAIC)*, 2001, pp. 283–290.

[26] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Comput. Vis. Graph. Image Process.*, vol. 29, no. 3, pp. 273–285, 1985.

[27] A. Rozinat and W. M. P. van der Aalst, "Conformance checking of processes based on monitoring real behavior," *Inf. Syst.*, vol. 33, pp. 64–95, Mar. 2008.

[28] Research Data. *Hospital Billing—Event Log*. [Online]. Available: https://data.4tu.nl/repository/uuid:76c46b83-c930-4798-a1c9-4be94dfeb741

[29] N. Tax, X. Lu, N. Sidorova, D. Fahland, and W. M. P. van der Aalst, "The imprecisions of precision measures in process mining," *Inf. Process. Lett.*, vol. 135, pp. 1–8, Jul. 2018.

[30] W. M. P. van der Aalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Berlin, Germany: Springer, 2011, p. 352. doi: 10.1007/978-3-642-19345-3.

[31] D. Chapela-Campa, M. Mucientes, and M. Lama, "Mining frequent patterns in process models," *Inf. Sci.*, vol. 472, pp. 235–257, Jan. 2019.

[32] D. Knoll, J. Waldmann, and G. Reinhart, "Developing an internal logistics ontology for process mining," *Procedia CIRP*, vol. 79, pp. 427–432, 2019. doi: 10.1016/j.procir.2019.02.116.