Computing and Informatics, Vol. 45, 2026, 231–260, doi: 10.31577/cai_2026_1_231

BEHAVIOUR ANONYMOUS METHOD OF BUSINESS PROCESS BASED ON LOG SKELETON

Xinsheng Fang, Xianwen Fang*, Ke Lu

School of Mathematics and Big Data
Anhui University of Science and Technology
Huainan 232001, Anhui, China
e-mail: {2022100082, xwfang, kelu}@aust.edu.cn

Abstract. Privacy-preserving process mining (PPPM) is a technology that analyses and optimises processes while safeguarding sensitive information. Earlier research on PPPM mainly focused on the data perspective, employing techniques such as noise insertion and data generalisation to protect the sensitive personal information of process executors. However, these studies overlooked the behavioural relationships between activities in the process. Attackers can exploit domain knowledge and certain behavioural information of executors to carry out re-identification attacks, leading to the leakage of personal sensitive information. To address this issue, a behaviour anonymous method of business process based on a log skeleton is proposed. This method begins with the individual process model of the executor, clustering based on the similarity between models and utilising K-anonymity and the log skeleton technology to achieve cluster division and the construction of behaviour constraint sets. Furthermore, the generation of privacy models is standardised by the behaviour constraint set to achieve global behaviour anonymization. To evaluate the effectiveness of this method, experiments were conducted using multiple real and synthetic datasets. The experimental results indicate that this method significantly outperforms the comparison methods.

Keywords: Privacy protection process mining, behavioural anonymization, log skeleton, behavioural relationships, behavioural constraint set

Mathematics Subject Classification 2010: 68-Q01

* Corresponding author

1 INTRODUCTION

Information systems are rapidly evolving and the event logs they generate can be used for comprehensive analysis of business processes [1]. These event logs are the basis for model improvement and process optimisation. Still, the logs contain a large amount of personal privacy information of process executives. Business units will not quickly provide the logs to third-party process analysts, resulting in obstacles to the optimisation and analysis of the process. So, the study of process mining privacy protection techniques has received extensive attention in recent years. Felix Mannhardt et al. [2] addressed the problem of applying PPPM techniques in specific environments to achieve a more contextually appropriate process optimisation solution for process executives while ensuring individual privacy. Elkoumy et al. [3] generalised and summarised the problems to be solved by the existing PPPM, and, for the actual medical scenarios, they formalised several privacy attack models. They presented the privacy challenges the business process domain faced, pointing the way for subsequent related research. Elkoumy et al. [4] proposed a differential privacy release mechanism for the log anonymization problem, which uses a sampling method to inject noise into process traces without altering the overall trace distribution. According to this method, the processed logs' timestamps vary within a prescribed range, minimising the loss of utility of the logs. Still, the technique ignores organisational provisions to anonymise all timestamps. Meanwhile, Elkoumy and Dumas [5] proposed a sampling-based noise insertion method for better protection of log utility; this method first filters out the trace variants that are prone to privacy leakage and then extracts multiple traces from the filtered logs for noise insertion to obtain anonymised privacy logs, the evaluated event logs can be assessed to provide optimal privacy guarantee, but the method does not apply to event logs with the presence of low-frequency valid trace variants.

Current business process privacy research mainly focuses on the data attribute perspective, Batista and Solanas [6] proposed a PPPM method based on event distribution unification for location and finite space identification based attack patterns, which focuses on the link between process executor's identity information and data attributes, and achieves privacy preservation of the process data by using the cluster unification method approach. Rafiei and van der Aalst [7] formally define the main privacy anonymization protection methods in order to protect the usability of the process data after anonymization, and develop a data standard for nontraditional event log data that guarantees the analytics of the data in any form after anonymization and also provide a library of packages that can be used to support process mining privacy preservation work. Rafiei et al. [8] proposed a privacy-preserving model for process event logs based on the LKC privacy model, which utilised the group's anonymity-preserving approach to anonymise log data and proved experimentally that the approach could maintain high process analytics while guaranteeing privacy usability. Pika et al. [9] evaluated the applicability of existing anonymization methods for process data with respect to the privacy needs of the healthcare domain and proposed a process mining privacy protection framework which analyzes and trans-

forms healthcare process data based on privacy metadata to achieve process privacy protection. Mannhardt et al. [10] analysed event logs in terms of the path's privacy needs and guaranteed the utility of the logs after privacy processing, proposed a process mining privacy preservation technique that combines differential privacy and data transfer mechanisms to add noise to the query results of non-trusted third parties to ensure that the anonymization of sensitive information about the process executors is achieved.

Some of these researchers have investigated process data privacy protection in a multi-organisational process interaction environment. Liu et al. [11] proposed a cross-organizational process mining privacy preservation approach in order to solve the cross-organizational business process privacy preservation problem, which focuses on data-centric privacy preservation in event logs, and provides interoperability solutions for multi-organizations while safeguarding the privacy of private data. Elkoumy et al. [12], to achieve multi-organisational process analysis and intra-organisational data privacy protection, proposed a secure multi-party process mining approach which achieves the privacy protection of internal and public data of all parties through segmentation sharing and encrypted transmission techniques to provide usable solutions for multi-organisational process data processing.

However, event logs exist for multi-dimensional information, of which behaviour is one of the crucial dimensions [13]. If behavioural relationships are not considered during the introduction of noise, an attacker can quickly identify that the published information has been processed for privacy; to achieve privacy while considering behavioural impacts, Fahrenkrog-Petersen et al. [14] proposed two trace variants anonymization methods, the former focuses on the process semantics of the flow to achieve reasonable insertion of noise, but this method changes the original trace variant distribution, to solve this problem the latter achieves anonymization based on direct follow relationships between activities and replay techniques to get a trace variant distribution more similar to the original data. Batista et al. [15], to solve the problem of location-based attacks in public environments, proposed a micro-aggregated business process privacy-preserving approach, which utilises the concept of K-anonymity to achieve anonymization of process logs, and the anonymised logs can be used to achieve multiple individual process representations using a single process model. Rösel et al. [16], to avoid unsemantic activities during the anonymization process, proposed a feature learning-based distance metric based on embedded methods, capturing contextual information about the activities to achieve a semantic distance metric between traces. At the same time, the technique guides the merging of traces with utility maximisation to achieve anonymization of the event logs.

Existing work on process mining privacy protection mainly focuses on the data attribute perspective. Most of the studies study the possible privacy leakage problem caused by business process event logs from the dimension of trace variants, ignoring the impact of inter-activity behavioural relationships on privacy disclosure, so this paper proposes a behavioural constraints-based business process anonymization approach, which starts from the similarity of the behavioural model of the process

executives to achieve the anonymization of business process behaviours. The specific contributions of this paper are as follows:

1. To implement the generation of privacy traces based on inter-activity behavioural relations, extracting inter-activity behavioural relations from incomplete clusters to construct behavioural constraint sets, thereby guiding prefix expansion and generating privacy traces with similar behaviours.
2. Propose a privacy model filling-based process behaviour anonymization method to screen the generated privacy traces and construct a privacy model to achieve process behaviour anonymization by adding privacy traces to incomplete clusters.
3. Focusing on the behavioural perspective of privacy preservation, business process behaviour anonymization is achieved by generating K process traces with similar behaviours, avoiding the risk of re-identifying individual executors.

The rest of the paper is organized as follows: Section 2 gives a motivating case for the problem solved in this paper, detailing the impact of behaviour on process privacy preservation. Section 3 will introduce some of the foundational concepts used in this paper. Section 4 is the main body of the paper, which mainly describes the concrete implementation of the proposed method in this paper; Section 5 evaluates the proposed method and verifies its validity of the proposed method, and Section 6 sums up the whole paper and discusses the future directions of the research.

2 MOTIVATION

Currently, the research on business process privacy protection mainly focuses on the sensitive data information of process executors, and reduces the difference between sensitive data and ordinary data through distribution unification, data transformation, differential privacy, etc., in order to realize the anonymization of business process data. Some studies focus on the privacy disclosure problem that exists during multi-organizational interactions, and achieve privacy support during multi-organizational information interactions in the form of segmentation sharing, encrypted transmission, and secure interoperability schemes. There are also a few studies that focus on the privacy disclosure problem that may be caused by the control flow hierarchy in a process, and anonymization of the control flow hierarchy is achieved through feature learning and clustering methods.

These studies mentioned above mainly use event logs as an entry point to minimize the risk of process executor information leakage through methods such as anonymization and noise insertion. The anonymization method circumvents the identification risk by hiding the executor's attribute information, but the attacker can construct a comparison model based on the corresponding domain knowledge and some of the executor's behavioural information obtained through observation, and then mine the entity model by organizing the log information released to achieve

the re-identification of the executor's information based on model similarity comparison. Table 1 shows a process log fragment for hospital detection, which mainly contains 2 cases and 12 activities. For the example fragment, assuming that the attacker knows the patient's identity information, he or she can infer the patient's disease information based on some of the patient's behaviours in the hospital environment, e.g., when the attacker learns that the patient has consecutively performed the Abdominal B-ultrasonography and Renal function test activity, he or she can identify the patient's diagnosis information by comparing the cases 19, which can lead to the disclosure of the patient's privacy.

Event ID	Case ID	Activity Name	Timestamp
...	...	...	...
88	19	Register	2022-09-02 09:11:08
89	19	Abdominal b-ultrasonography	2022-09-02 09:30:22
90	19	Renal function test	2022-09-02 10:11:08
91	19	Urine routines	2022-09-02 10:45:39
92	19	Prescribe	2022-09-02 11:05:10
93	20	Register	2022-09-02 08:30:39
94	20	Blood routine examination	2022-09-02 09:10:25
95	20	Abdominal b-ultrasonography	2022-09-02 09:50:12
96	20	Liver function test	2022-09-02 10:20:21
97	20	Prescribe	2022-09-02 10:50:45
...	...	...	...

Table 1. Example of an event log

The noise insertion approach to privacy processing from the dimension of event traces ignores the behavioural relationship between activities, which may lead to the generation of a large number of unreasonable traces; for example, in the example fragment, the starting activity of each case is Register, and the final activity is Prescribe, if the noise insertion approach is used to avoid the re-identification and speculation attacks, it may drive some cases to have an initial activity that is not Register or to have Prescribe activity immediately after the Register activity, which may lead to the attacker being able to identify quickly that the log has been processed for privacy.

Different from the traditional log clustering method, this paper first builds a behavioural model based on the individual process traces of the executors, and clusters them by the similarity between the models. Secondly, incomplete log clusters are separated according to privacy parameters. Then, in order to analyse the impact of behavioural relationships between activities on personal information disclosure, the log skeleton method is used to extract the behavioural relationships between activities in the incomplete clusters and construct a behavioural constraint set. Finally, based on the behavioural constraint set and all possible prefixes in the global scope, similar traces are generated, and the generated traces are screened according to the violation counts. A privacy model is built based on the screened generated

traces, and these models are added to the incomplete clusters, thereby achieving global behavioural anonymity.

3 BACKGROUND

This section will provide some background on traces, events, event logs, K-anonymity, and more.

Definition 1 (Event [17])**.** The occurrence of a series of events constitutes a business process, where each event can be denoted as $e = (caseid, a, time, resource)$ *caseid* denotes the case identifier corresponding to the event, a denotes the activity corresponding to the event, *time* denotes the time of execution of the activity, and *resource* usually denotes the executor of the activity or the object that accomplishes the operation. An event can carry a variety of attribute information such as the name of the activity, the execution time of the activity, the executor of the activity, and the executor's information.

Definition 2 (Trace, event log [18])**.** A trace is a non-empty sequence of multiple events executed sequentially, which can be expressed as $\sigma = e_1e_2e_3e_4e_5 \ldots .e_n$ $(n \geq 1)$, where the time of occurrence of two neighbouring events within the trace, the former always has to be less than or equal to the latter, i.e. $time\,(e_i) \leq time\,(e_{i+1})$ and $1 \leq i \leq n$.

Similarly, the event log is a collection of multiple traces, which can be expressed as $\log = \{\sigma_i \in s \mid 1 \leq i \leq m\}$, representing the set of all possible traces and the number of traces contained in the event log.

As shown in Table 1, the event log segment, which consists of four main attributes: Event ID, Case ID, Activity Name, and Timestamp, there are two cases (19 and 20) as well as ten activities.

Definition 3 (Prefix traces [19])**.** The prefix of a trace is a subsequence from its initial activity and the set of trace prefixes contains the trace itself. For a trace $\sigma =< e_1, e_2, \ldots, e_n >$, its prefix is $\sigma_l = \langle e_1, e_2, \ldots, e_l \rangle$, where $1 \leq l \leq n$.

Definition 4 (K-anonymity [20])**.** In the data table T, if any A^{qi} in pattern $R(A_1, A_2, \ldots, A_n)$ always satisfies $A_i^{qi} \times \mathrm{t}\,(\mathrm{A}_s^{qi}) \geq K$, then we consider the data table to satisfy K-anonymity.

Where K represents the degree of anonymization, it is required that $K-1$ individuals exist similar to all the data's personal information, such that the probability of the individual being identified is only $1/K$.

This paper defines the concepts of behavioural K-anonymity and incomplete clusters to apply K-anonymity to the business process behavioural perspective.

Definition 5 (Behavioural K-anonymity)**.** In the model library M, there are always at least $K-1$ models in any model M_i in pattern $R(M_1, M_2, \ldots, M_N)$ that

exhibit similar behaviour, expressed as:

$$|R(M_i)| \forall M_i \in M \mid \geq K. \tag{1}$$

Definition 6 (Incomplete Cluster)**.** Let L be an event log, C_n be a cluster in the clustering, and for the privacy parameter K, $\sigma_n \in L$ be all possible traces contained in cluster C_n. If $\mid \sigma_n \mid < K$, then we call the cluster an incomplete cluster.

Definition 7 (Log Skeleton [21])**.** Let L be an event log, $\sigma \in L$ denotes a trace in the event log, the log skeleton $SK(L)$ mainly contains five behavioural relations respectively Eq, Aa, Ab, Nt, Df where:

- Eq denotes an equivalence relation:

$$((a,b) \in Eq) \Leftrightarrow (\forall_{\sigma \in L} \mid \sigma * \{a\} \mid = \mid \sigma * \{b\} \mid), \tag{2}$$

  a, b are two activities which, if they have behavioural equivalence, imply that they occur with the same frequency in each trace, where $\mid \sigma * \{a\} \mid$ is used to denote the frequency of occurrence of activity a in trace σ.

- Aa denotes an always-follow relationship:

$$((a,b) \in Aa) \Leftrightarrow (\forall_{\sigma \in L} (\sigma * \{a\} = \langle\rangle) \vee (\nabla (\sigma * \{a,b\}) = b)), \tag{3}$$

  a, b has an always-follow relationship, for all traces in the event log, if $\{a,b\}$ exists in trace σ and when activity a occurs, activity b always occurs, then we say that b always follows a, and the two are in an always-follow relationship, and in the expression ∇ denotes the last activity.

- Ab denotes an always-before relationship:

$$((a,b) \in Ab) \Leftrightarrow (\forall_{\sigma \in L} (\sigma * \{a\} = \langle\rangle) \vee (\Delta (\sigma * \{a,b\}) = b)), \tag{4}$$

  a, b has an always-before relationship. For all traces in the event log, we say that they have an always-before relationship if $\{a,b\}$ exists in trace σ and activity a always occurs when activity b occurs, with Δ denoting the first activity in the expression.

- Nt denotes a never co-occurrence relationship:

$$((a,b) \in Nt) \Leftrightarrow (\forall_{\sigma \in L} (\sigma * \{a\} = \langle\rangle) \vee (\sigma * \{b\} = \langle\rangle)), \tag{5}$$

  a, b has an never-co-occurrence relationship. For all traces in the event log, if there is activity a, there is no activity b, and if there is activity b, there is no activity a, then we say that they have a never-co-occurrence relationship.

- Df denotes a direct-follow relationship:

$$((a,b) \in Df) \Leftrightarrow \left(\underset{\sigma \in L}{\exists} \sigma \diamond \langle a,b \rangle > 0 \right), \tag{6}$$

a, b has a directly-follow relationship, for all traces in the event log, we say that they are in a direct-follow relationship if $\{a, b\}$ exists in the trace σ if and only if activity b occurs after activity a, and $\diamond$ denotes the number of occurrences of the activity in the expression.

4 METHOD

Existing research on privacy protection mainly explores the data flow perspective, such as protection methods based on data anonymity, distribution unification methods based on sensitive attributes, and data protection methods based on differential privacy, etc., in which traditional data anonymity achieves privacy protection through data generalisation and data pseudonymisation, which leads to a significant reduction in the usability of the process information, and makes subsequent process optimisation efforts become extremely difficult; while the method of changing data distribution leads to the mutation of the original behavioural details of the executor, and if the attacker has domain knowledge, the attack can be quickly completed; and finally, the method about differential privacy, the differential privacy technique can effectively prevent inference attacks through fine-grained perturbation insertion, but the application of this method in the field of process mining has certain defects, and may construct the original behavioural relationship which does not conform to the original traces of behavioural relations.

To protect the executor's data information from disclosure due to the behavioural disclosure problem, an encryption method that can implement the executor's behavioural relationship is needed. So, a privacy trace generation method based on behavioural constraints is proposed to achieve the K-anonymity of the executor's behavioural model through clustering ideas.

In Figure 1, the general framework diagram of this paper is shown. Firstly, starting from the event logs, the process traces are split based on the execution individuals in the logs, secondly, the process models are constructed for different individuals, and based on the similarity between the models, using clustering method, all the behavioural models are aggregated into multiple clusters, and then the clusters are divided based on whether they satisfy the privacy parameters or not, and for the incomplete clusters, the inter-activities are extracted based on log skeleton methods behavioural relationships and define the set of behavioural constraints, generate privacy traces with similar behaviours to other models in the incomplete clusters under the condition of behavioural constraints, and filter the generated traces, and finally use the filtered traces to construct the privacy models and add these models to the incomplete clusters until they satisfy the privacy requirements, so as to achieve the K-anonymity of the overall behaviours.

Different from traditional process mining privacy protection techniques, the method is based on the premise of the behavioural relationship between activities; there will not be an unrealistic order of occurrence of activities; for example,

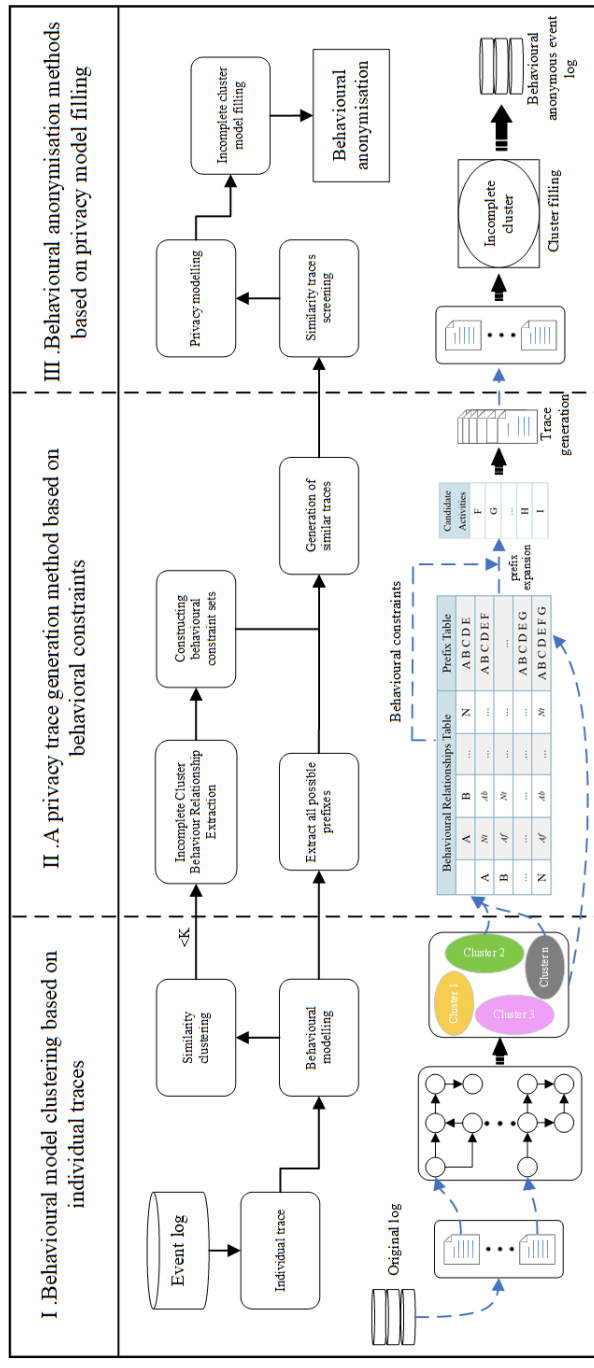


Figure 1. General framework

in the process of medical diagnosis, the patient needs to register before the corresponding examination, if the other examination process appears before the registered activity, obviously does not conform to the basic common sense, the attacker can quickly determine the release of the information encrypted processing, at the same time The presence of such traces in a completely different order of occurrence from the original logs can have a significant impact on the work of process analysts.

4.1 Behavioural Model Clustering Based on Individual Traces

In the traditional privacy protection domain, data is always the priority. Still, in the process mining domain, the actual behaviours of the process executors are an essential aspect that cannot be ignored. In a public environment, the various data metrics of the process executor may be released in encrypted form. Still, the behavioural information executed in that environment is available. The attacker can construct a model based on the relevant information released in the public environment and the observed behavioural information, respectively, and then make use of the similarity of the model to infer the actual process model of the attacked person, which will lead to the re-identification of the process executor, which will in turn to disclose sensitive information about their individuals. To protect the behavioural privacy of process performers and avoid this risk of re-identification, this paper constructs behavioural models based on individual process traces and achieves clustering based on the similarity between the models, which lays the foundation for the subsequent generation of privacy traces and the implementation of behavioural anonymity.

While traditional trace similarity calculation methods quantify elements by their frequency of occurrence or the number of identical elements, this section proposes a behavioural model clustering method based on individual traces that take the behavioural relationships between activities into account.

The method first starts from the event log L. Assuming that the event log L contains process traces of n executors, describing all the activities performed by the executors in the current environment, the log is divided into n sub-logs based on the criteria of the executors, and then the discovery algorithm is used to construct the individual behavioural model of each executor, which at this time contains the behavioural information of the executors as well as sensitive data information. Trace clustering, the similarity matrix, is constructed by comparing the similarity of individual behaviour models to guide the clustering of process traces. Different from previous clustering methods, we need to set an adjustable privacy parameter K to control the degree of privacy and achieve the classification of clusters based on the privacy parameter. The value of K is in the range of $[2, n]$, and the degree of confidentiality is highest when all the behaviours are in the same cluster. The detailed steps of the trace clustering method based on the individual behaviour model are shown in Algorithm 1.

Algorithm 1: Model clustering based on behavioral similarity

input : Privacy parameter K;An empty matrix $Ma_{n\times n}$; Number of individuals n; Clustering Method CM

output: Two collections containing clusters Cs_1,Cs_2

1 $Mn \leftarrow \emptyset$;
2 $Ln = List <>$;
3 **while** $0<i \leq n$ **do**
4 $\quad Ln_i = gettrace\,(i)$;
5 $\quad Mn_i = getmodel\,(Ln_i)$;
6 $\quad Ln = Ln.appen\,(Ln_i)$;
7 $\quad Mn = Mn.appen\,(Mn_i)$;
8 **end**
9 $Ma \leftarrow TheInitializationMatrix$;
10 **for** *0≤i≤n-1* **do**
11 $\quad$ **for** *i≤j≤n-1* **do**
12 $\quad\quad Sim \leftarrow$ Distance(Mn_i, Mn_j);
13 $\quad\quad$ Ma[i][j]=Sim;
14 $\quad\quad$ Ma[j][i]=Sim;
15 $\quad$ **end**
16 **end**
17 $C \leftarrow CM(Ma, K, Mn)$;
18 Cs_1,$Cs_2 \leftarrow$ Split(C);
19 return Cs_1,Cs_2

Algorithm 1 firstly inputs the privacy parameter K and the number of individuals n, and initialises M_n to represent the behavioural model of each performer (line 1), secondly, the event log is split to obtain the individual behavioural traces of each performer (lines 2–4), and the behavioural model is constructed using the obtained individual traces of the performers (lines 5–8), and then we compute the similarity values between the individual behavioural models, and use these values to construct the similarity matrix (the constructed similarity matrix has symmetry), and according to the constructed similarity matrix to achieve the clustering of the models (lines 9–16), and finally use the privacy parameter to classify all the clusters, if the number of models contained in the cluster is more than or equal to K clusters are classified as C_{s1}, and vice versa the number of models contained in the clusters is less than K are classified as C_{s2} (incomplete clusters), and finally output the classification results (lines 17–19).

Algorithm 1 is different from the traditional method of trace similarity comparison; to pay more attention to the behavioural information of the process, individual traces are used to construct a behavioural model, and clustering is achieved based on the behavioural model similarity, which can reasonably take the flow of execution of the activities, as well as the behavioural relationship between the activities into consideration, and pay attention to the spatial attributes of the process behaviours.

4.2 Behavioural Constraint Set Construction Method Based on Log Skeleton

In the previous section, we implemented the clustering of individual behavioural models of process executors and used privacy parameters to filter out incomplete clusters that do not meet the privacy requirements of the clustered clusters. In order to achieve the goal of global behavioural anonymization, in this paper, we make use of the filling of the privacy model to motivate the incomplete clusters to meet the privacy requirements. In order to keep the behaviour of the filled models consistent with the original models in the clusters, the log skeleton technique is used to extract the constraint relations between activity pairs from the incomplete clusters, and the obtained constraint relations are used to guide the generation of privacy models.

For the executor of each process trace in the log, the activity that occurs is single-threaded, from the beginning until the end, but when there are multiple processes in the scenario, there is a behavioural relationship between the activities such as always-before, never co-occurrence, etc., e.g. in the medical scenario mentioned above, the registration is certain to occur before the prescribing activity, and the two have a strict sequential relationship; and second, for the Abdominal b-ultrasonography and Color doppler imaging activities, both of them have a partial functional overlap, and some patients can get a diagnostic conclusion only by performing one of them. functionally overlap, some patients only need to perform one of the activities to get the diagnostic conclusion, so for the patient these two activities present a never-co-occurrence relationship. The log-skeleton approach is able to extract the constraint relationships between activities in a global view, focusing on all traces in the incomplete cluster. This section is mainly based on the log-skeleton method to extract five behavioural relations such as equivalence, always-before, never co-occurrence, etc. from incomplete clusters to construct the behavioural constraint set. The specific steps are shown in Algorithm 2.

Algorithm 2 firstly inputs C_{s2} set of incomplete clusters, initializes 5 behaviour relationships of the log skeleton and 2 parameters *Beh* and *logskeleton* (line 1–4), secondly for each incomplete cluster, extracts the behavioural relations between the activities from the flow traces it contains, calculates the frequency of occurrence of each activity (lines 5–7), then if the frequency of occurrence of the two activities is equal then add the activity into the set of equivalence relation constraints; if the two activities are always indexed before or after the index of the other activity, then the activities are added to the always-before or always-follow constraint set; for the never co-occurrence relationship, the index of one activity never occurs after the index of the other activity, then the activities are added to the never co-occurrence constraint set; if the two activities always happen before or after each other then the two activities are said to satisfy the direct-follow relationship, and the activities are added to the direct-follow constraint set (lines 8–23), and finally assign all behavioural constraints to and eliminate all duplicates to output the log skeleton behavioural constraint set *logskeleton* (lines 24–28).

Algorithm 2: Behavior Constraint Set Construction Based on Log Skeleton

input : Privacy parameter K;A cluster of clusters Sn_i;The set of clusters smaller than privacy parameter Cs_2
output: Set of behavioral constraints *logskeleton*

$Eq_\sigma, Aa_\sigma, Ab_\sigma, Nt_\sigma, Df_\sigma \leftarrow \emptyset$;
$Eq_\sigma, Aa_\sigma, Ab_\sigma, Nt_\sigma, Df_\sigma \epsilon$ *logskeleton*;
$Sn_i \epsilon Cs_2, Ln_i \epsilon Sn_i$;
$Beh, logskeleton = \emptyset$;
for $0 \leq i < |Ln_i|$ **do**
  for $i \leq j < |Ln_i|$ **do**
    $activitycounts$ = collections.Counter(A_i);
    if $activitycounts[A_i] == activitycounts[A_j]$ **then**
      $Eq_\sigma \leftarrow (A_i, A_j)$;
    end
    if $activities.index(A_i) < activities.index(A_j)$ **then**
      $Aa_\sigma \leftarrow (A_i, A_j)$;
    end
    if $activities.index(A_i) > activities.index(A_j)$ **then**
      $Ab_\sigma \leftarrow (A_i, A_j)$;
    end
    if *a not in activities*[*activities.index(b)+1:*] **then**
      $Nt_\sigma \leftarrow (A_i, A_j)$;
    end
    else $A_i = activities[i], A_j = activities[i+1]$
      $Df_\sigma \leftarrow (A_i, A_j)$;
    end
  end
  $Beh \leftarrow Eq_\sigma, Aa_\sigma, Ab_\sigma, Nt_\sigma, Df_\sigma$;
  $Beh = Beh.duplicated(Beh)$;
end
$logskeleton = logskeleton.append(Beh)$;
return *logskeleton*

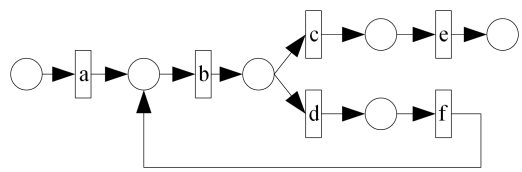


Figure 2. Petri net example 1

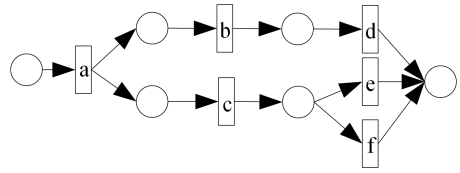


Figure 3. Petri net example 2

In the following, a concrete example is presented to illustrate the method of constructing the set of behavioural constraints, where the behavioural constraints are extracted from all possible traces of the two example Petri nets of Figures 2 and 3. Both Petri nets have six activities from a to f. The possible traces of Figure 2 are $\sigma_1 = [(abce)(abdfce)]$, then its behavioural constraints are: $Eq_\sigma = \{(c,e)(d,f)\}$; $Nt_\sigma = \{\emptyset\}$; $Aa_\sigma = \{(a,b),(c,e),(d,f)\}$; $Ab_\sigma = \{(e,c),(f,d),(c,b),(d,b)\}$. The possible traces of Figure 3 are $\sigma_2 = [(abd)(ace)(acf)(abcde)\dots]$, then their behavioural constraints are: $Eq_\sigma = \{\emptyset\}$; $Nt_\sigma = \{e,f\}$; $Aa_\sigma = \{(a,b),(a,c),(b,d)\}$; $Ab_\sigma = \{(b,a),(c,a),(d,b),(e,c),(f,c)\}$; $Df_\sigma = \{(b,d)\}$.

4.3 Privacy Trace Generation and Behavioural Anonymization Implementation

In the previous two sections, we filter incomplete clusters based on privacy parameters and construct a behavioural constraint set based on the log skeleton method. To achieve the global behavioural anonymization goal, it is necessary to generate privacy traces with behaviours similar to those in the current incomplete clusters under behavioural constraints and construct a privacy model that fills in the current incomplete clusters until they meet the privacy requirements. Thus, a privacy trace generation method based on behavioural constraints and a behavioural anonymization method based on privacy model filling is proposed. Firstly, all possible prefixes are extracted in the global scope class. Secondly, all possible subsequent activities are extracted, and the current prefix is extended by traversing to pick the next activity that meets the behavioural constraints until the end activity of the trace or until it reaches the maximum length. In this expansion process, since the selection of prefixes and following activity activities is global in scope, there may be behavioural relationships between activities that violate the current cluster so that a violation count will be defined below for privacy trace screening.

Definition 8 (Violation Count (Vic))**.** Assume that there exists a sequence $\mid \sigma = \langle a_1, a_2, \dots, a_i \rangle \mid \in \sigma^*$, σ^* is the set of activity occurrence sequences in log L. The violation count Vic is increased by 1 if $R(a_i, a_{i+1}) \notin logskeleton$, and the equivalence relation is not taken into account in the calculation of the violation count.

This violation count measures the generated candidate privacy traces by the set of behavioural constraints extracted from incomplete clusters, and since the equiv-

alence relation in the behavioural constraints does not necessarily affect the generation of behavioural similar traces, the proposed violation count does not consider the equivalence relation. After obtaining the violation counts of the privacy traces generated based on the behavioural constraints of the current incomplete cluster, the violation counts are used to filter the candidate traces. The privacy traces with relatively small violation counts are selected to construct a privacy model to fill in the incomplete cluster and iterate the current step until the global behavioural anonymization is achieved, as shown in the specific steps in Algorithm 3.

Algorithm 3: Behavior Anonymization Method Based on Privacy Model Filling

input : Initial activity A_x;The set of clusters Cs_1,Cs_2;Privacy parameter k
output: K-anonymous clustering C

1 $H \leftarrow \emptyset$;
2 $Sn_i \leftarrow Trace(Cs_1 + Cs_2)$;
3 $MP = \text{Max}(|Ln_i|, Ln_i \epsilon\ Sn_i)$;
4 $G = Getactive(Ln_i)$;
5 $GTP = Gettraceprefix(Ln_i)$;
6 $TP = List <>$;
7 **for** *every* $Sn_j \wedge Sn_j\ \epsilon\ Cs_2$ **do**
8 $n_k = k - N_{Sn_i}$;
9 **for** $x = 1 : f(nk)$ **do**
10 $TP \leftarrow range(GTP)$;
11 **while** $|TP| < MP \wedge TP[-1] \neq e$ **do**
12 $y = 0 : |G(Ln_i)|$;
13 $A_y \leftarrow [G(Ln_i) \cup e][y]$;
14 $A_x \leftarrow TP$;
15 **if** $< A_x, A_y > \epsilon\ logskeleton$ **then**
16 $TP = TP.append < A_y >$
17 **end**
18 **end**
19 $H = H.appen(TP)$
20 **end**
21 $STP = select(Vic(H), nk)$;
22 $Sn_j \leftarrow add(getmodel(STP))$;
23 **end**
24 return $C = Cs_1 + Cs_2$

Algorithm 3 firstly inputs the privacy parameter K and two sets C_{s1}, C_{s2} which are classified according to the value of K. Secondly, it initialises the candidate set of privacy traces (line 1), extracts all the possible trace prefixes globally and assigns the maximum length of the trace to the MP (lines 2–5), and then declares an empty list for storing the prefix extensions (line 6), and for all the possible prefixes, the next activity to implement the prefix is selected from all the potential candidates, under the guidance of the set of behavioural constraints, until the maximum

length is reached or the termination symbol 'e' (lines 7–18), to get the candidate set of privacy traces. Under the guidance of the behavioural constraint set, the next activity from all possible candidate activities is selected to achieve the extension of the prefix until it reaches the maximum length or the termination symbol 'e' (lines 7–18) to obtain the candidate set of privacy traces H (line 19), and then the traces are screened according to the violation count, and the screened privacy traces are used to construct the behavioural model to fill in the incomplete clusters to satisfy the privacy requirements, and finally output event logs that satisfy behavioural anonymity (lines 21–24).

To represent the generation process of privacy trace more formally, a concrete example is proposed in the following, as shown in Figure 4, there are several prefixes $\langle ABCD\rangle$, $\langle ABCE\rangle$, $\langle ACDE\rangle$ in the figure, and there are three possible follow-up activities, E, F, and G. The behavioural constraints assumed so far are that there exists a direct-follow relationship for activity AB and an always-follow relationship for activity CD. Then, it can be figured out that the cost of Prefix 1 is 0, and the cost of Prefix 2 and Prefix 3 are both 1. So, to get a trace with a better violation count, prefix expansion should be done based on Prefix 1, and the final possible trace is $\langle ABCDE\rangle$.

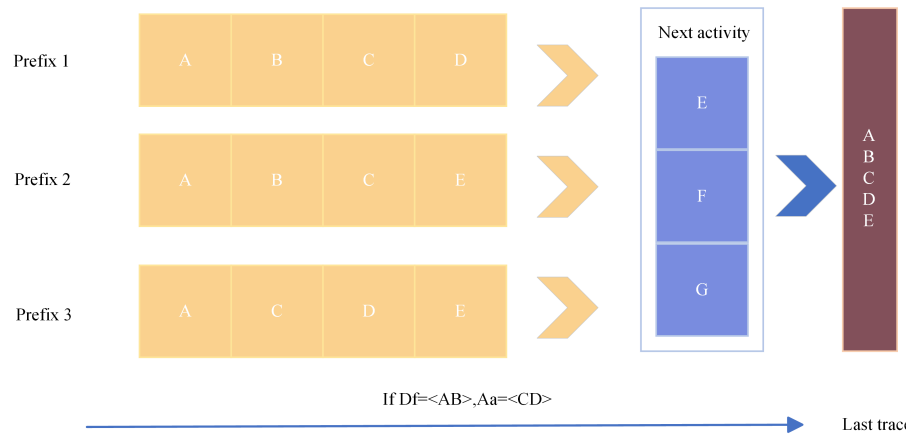


Figure 4. Schematic diagram of trace generation

The figure shows an example of the process. In the trace generation process, the complete behavioural constraint relationship must be considered to get a private trace that is more compatible with the current cluster behavioural constraints. There may be cases of multiple starting events in a massive event log, but the fine-grained generation method still follows the proposed approach.

This chapter describes the proposed method in detail. Unlike the generalised sampling approach, to guarantee the utility of the business process data and reduce the impact on the original data, the proposed method constructs behavioural

constraints based on the behavioural relationships between activities in incomplete clusters, which guides the generation of behaviourally similar privacy traces and the construction of a privacy model. Then, it uses a model-filling method to achieve the K-anonymity of the overall behaviours, which meets the privacy requirements while safeguarding the utility of the event log.

4.4 Complexity Analysis

To analyse the complexity of the proposed method, we will first pre-process the logs to remove events containing missing information, and its complexity is $O(E)$, where E is the total number of events; then for the construction of the individual process model of the executor, its complexity is $O\left(m \cdot |T|^2\right)$, where m denotes the number of traces in the event log, and $|T|$ denotes the number of activities per trace, and since the direct successor relationship is considered in constructing the individual process model , so its complexity is $O\left(|T|^2\right)$; further is the comparison of similarity between process models mainly using activity pair intersection and concatenation ratios, and its complexity is $O(|Mod|)$, and since there are m models (same as the number of traces), the complexity is $O\left(m^2 \cdot |Mod|\right)$; the complexity of clustering operation is $O\left(m^2\right)$, and finally is the checking of compliance with anonymity as well as model padding, and the complexity is $O\left(C \cdot K \cdot |T|^2\right)$, where C is the number of clusters and K is the privacy parameter, and assuming that each cluster has c models, the complexity of extracting behavioural relationships is $O\left(c \cdot |T|^2\right)$. We add up the complexity of each step to get the total complexity as $O\left(C \cdot K \cdot |T|^2\right) + O\left(m^2 \cdot |Mod|\right) + O(n)$.

5 EVALUATION

In this section, the proposed method will be evaluated in detail. First, the relevant setup regarding the evaluation and the equipment environment in which the experiments were carried out will be described. Second, the publicly available dataset and the synthetic dataset used will be described, and finally, the results obtained by the proposed method and the comparison method will be discussed.

5.1 Setup

To validate the effectiveness of the proposed method, it is evaluated from two aspects: firstly, to assess the behavioural changes of the output event logs after the privacy treatment, based on the event log mining process model, and then the original event logs are used to compute Fitness to measure the degree of behavioural difference between the original event logs and the event logs after the privacy treatment; secondly, to evaluate the effect of the privacy treatment before and after the privacy treatment on the utility of event logs, using the output logs for next activity prediction under different privacy parameters, and then comparing the obtained F1-

score [22], Precision [23], Recall [24] to analyse the impact of the proposed method on the utility of event logs.

In verifying the behavioural changes in the event logs after privacy processing, a heuristic-based approach is used to mine the Petri model from the event logs after privacy processing, and then the fit of the original logs to this model is calculated; in measuring the utility impact of the event logs, the next activity prediction is performed by training an extended short-term memory network (LSTM), taking the first 80% of the logs as the training set and the last 20% as the test set.

The experimental evaluation mainly contains the method proposed in this paper and two comparison methods; one of the comparison methods is the traditional event log K-anonymity [25] method, which is based on the frequency of occurrence of traces and filters the traces whose frequency meets the privacy requirements as the output; the other comparison method is the behavioural anonymization method based on sampling generalisation, which refers to the idea of sampling generalisation in the work of microaggregation [15], specifically, it randomly samples a certain proportion of logs corresponding to representative models from incomplete clusters for replicated generalization to achieve overall behavioral anonymization. The comparison of the three methods verifies the advantages of the proposed method in this paper on process mining behavioural privacy protection.

5.2 Data Set

The experiments mainly use three publicly available datasets CoSeLoG project [26], BPIC2020 [27], Sepsis [28] and a synthetic dataset, Synthetic data, where the CoSeLoG project dataset records the data of the building permit application process of the municipalities of several cities in the Netherlands, and since the activity labels of the permit application process perform the same activity labels, the process data from several cities together form this dataset, the BPIC2020 dataset is a travel reimbursement process log which contains both national and international category data, and the Sepsis dataset is a natural process log about hospitals detecting sepsis which includes several different trace variants. Synthetic data is an artificial event log that mainly describes the possible process activities in a healthcare scenario, from registration to prescribing medication or hospitalisation; the specific information of the dataset is shown in Table 2.

Event Log	Activity Number	Event Number	Case Number	Average Event Number
CoSeLoG project	27	8 577	1 434	6
BPIC 2020	19	72 151	6 449	5
Sepsis	16	15 214	1 050	14
Synthetic data	8	1 933	500	4

Table 2. Dataset information

As the real dataset is relatively large, the following synthetic dataset is used to demonstrate the extraction of the log skeleton behavioural relationships. Table 3 marks the five behavioural relationships as corresponding letters, while the case where the activities do not have a direct interrelationship with each other needs to be considered, where it is marked with o.

Original Name	Mark
Equivalence	e
Always after	a
Always before	b
Never co-occurrence	n
Direct follow	d
No relation	o

Table 3. Behavioural relationship mapping table

As shown in Table 4, the log skeleton behavioural relationship table extracted from the synthetic logs, due to the relative simplicity of the artificial logs, some of the relationships are not reflected in the actual behavioural extraction process if the two activities are direct-follow in fact, there is an equivalence relationship between the two, but in the process of generating the similar traces usually do not take into account the equivalence relationship between the activities, but only consider the direct follow relationship. To verify the reasonableness of the behaviour extraction, a heuristic mining method is used to mine the process model from the synthetic event log, and the results are shown in Figure 5, which shows that the behavioural relationship between the extracted activities is in line with the mined model, verifying the effectiveness of the behaviour extraction method.

Activity	Reg.	Abd.	Che.	Blo.	Uni.	Liv.	Hos.	Pre.
Reg.	n	b	b	b	b	b	b	b
Abd.	o	n	n	o	o	o	o	o
Che.	o	n	n	o	o	b	o	o
Blo.	o	o	o	n	n	n	o	o
Uni.	o	o	o	n	n	n	o	o
Liv.	o	o	o	n	n	n	o	o
Hos.	o	o	o	o	o	o	n	n
Pre.	o	o	o	o	o	o	n	n

Table 4. Synthetic log behaviour relationship table

5.3 Result Analysis

This section focuses on analysing the experimental results to validate the behavioural fit of the event logs after the privacy treatment and the impact that the privacy treatment has on the quality of the event logs. Since two different clustering methods

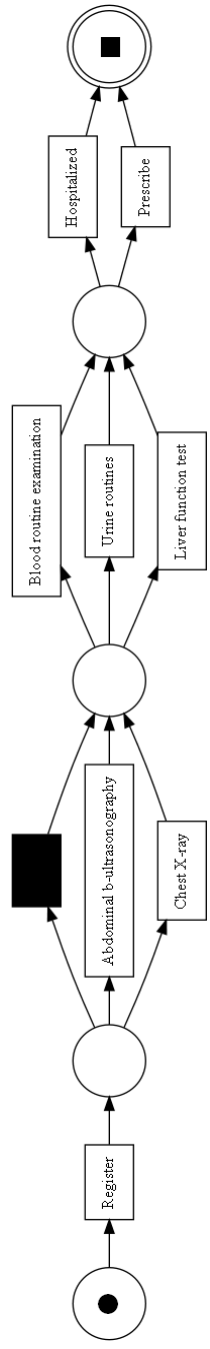


Figure 5. Synthetic log Petri net model

(DBSCAN [29], OPTICS [30]) were used in this paper for model clustering during the experiments, the results obtained from the two different clustering methods are also analysed during the evaluation process to assess the differences in the results produced under different clustering methods.

As shown in Figure 6, Fitness is used to measure the impact of the privacy treatment on the event logs' behaviour and the change in behaviour of the event logs before and after the treatment. In Figure 6, the horizontal axis represents the different privacy parameters, i.e., K values, and the vertical axis represents the fitness obtained by the two control groups (K-anonymous for the traditional K-anonymity method and G-anonymous for the sampling generalisation method) as well as the proposed method (B-anonymous) with different privacy parameters.

Figures 6 a), 6 b), 6 c), and 6 d) show the performance of the three methods on various datasets; in Figures 6 a), 6 b), and 6 c), it can be seen that the proposed method significantly outperforms the traditional K-anonymous method, and the fit of the conventional K-anonymous method shows a downward trend with the increasing privacy parameters because the traditional K-anonymous method keeps filtering out the trace variants whose frequency of occurrence is less than the privacy parameter. The sampling generalisation method always maintains a relatively high fit in these three result plots because the sampling generalisation method directly copies a certain percentage of the original traces for generalisation to achieve global anonymization, which means the behaviour of the privacy processed does not produce too much change, of course, there are certain defects in this method is easy to expand the noise contained in the original logs and change the original distribution of the data, resulting in the decline of the log quality. The method in this paper can overcome this problem, and the fit of the proposed method on the four datasets is not very different from that of the sampling generalisation method.

As shown in Figure 6 d), the results of the proposed method on the synthetic dataset are the same as the generalised sampling method, and the K-anonymous method is disregarded in this figure because the variability between the traces is too small.

Figures 7 a) and 7 b) show the results obtained by the proposed method using DBSCAN and OPTICS as clustering methods on the BPIC 2020 and Sepsis datasets, respectively. The results show that the choice of clustering method has a significant impact on the results, so when using the proposed method, it is necessary to select the appropriate clustering method based on the distributional characteristics of the event logs.

To measure the change in the quality of the event logs after privacy processing, the privacy logs obtained with different privacy parameters are used for the next activity prediction, and the impact of varying anonymization methods on the quality of the logs is evaluated by analysing the Precision, Recall, and F1-score metrics. In the following evaluation, the main comparison is between sampling generalisation and the proposed method; the K-anonymous method is not considered; this

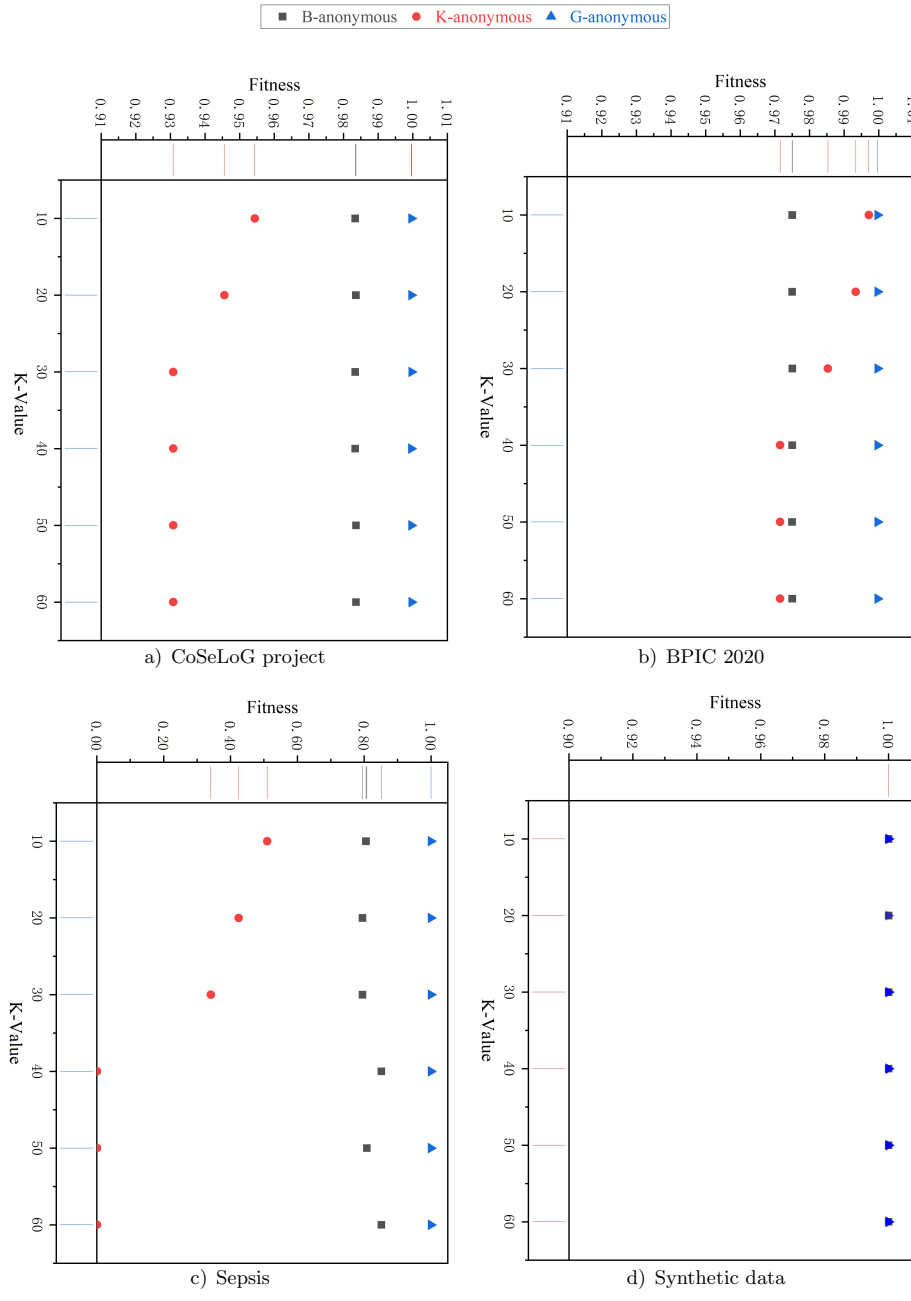


Figure 6. Fitness of different methods

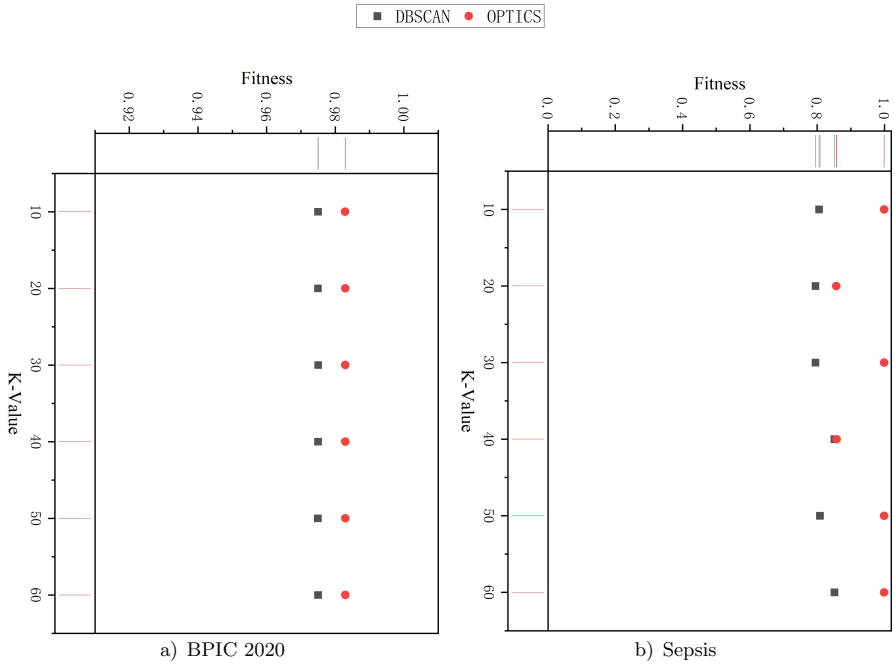
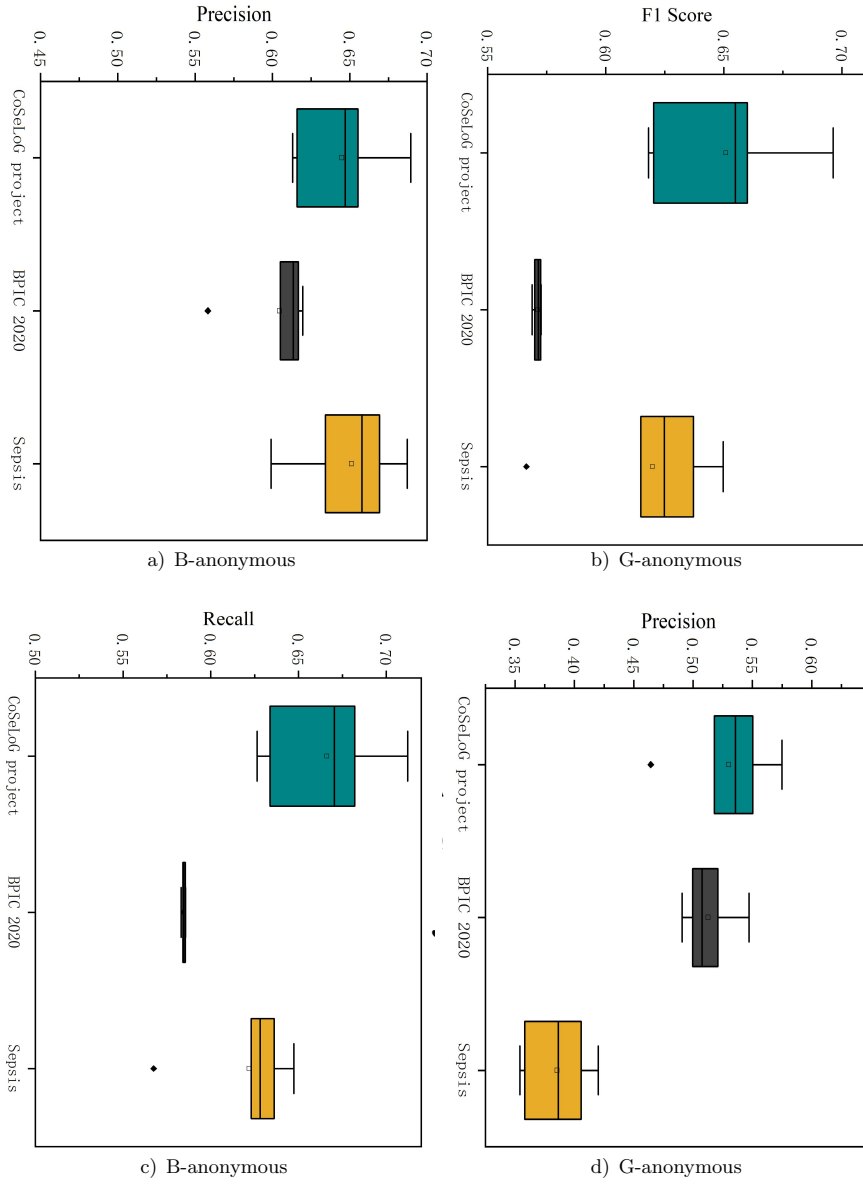


Figure 7. Fitness of different clustering methods

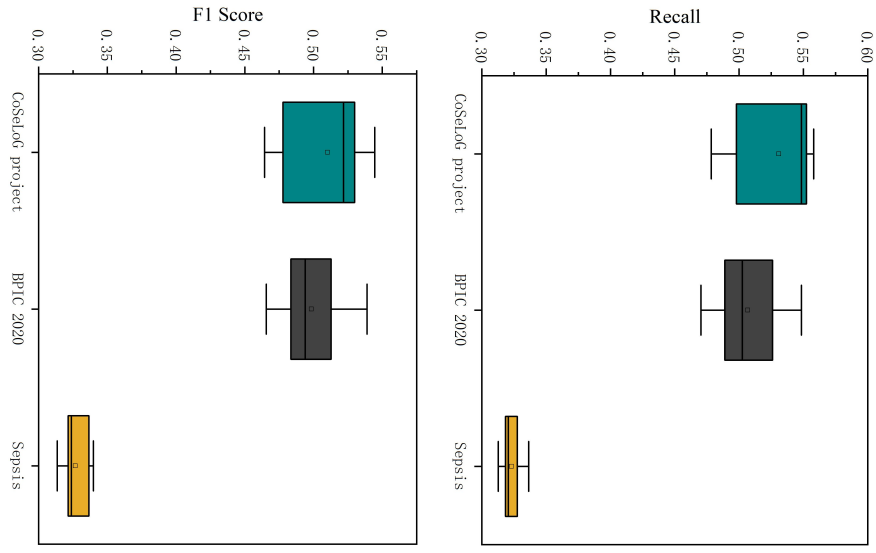
is because the conventional K-anonymous method constantly filters out the process traces that do not satisfy the privacy requirements, leading to a decreasing data size, which is not suitable to be used for the training of prediction models. Similarly, due to the amount of data, the synthetic dataset is not evaluated for the next campaign prediction.

Figures 8 a), 8 b), 8 c), 8 d), 8 e), and 8 f) show the Precision, F1-score, and Recall result plots obtained by the B-anonymous and G-anonymous methods on three publicly available datasets using the DBSCAN clustering method. It can be observed that on all three evaluation metrics, the method proposed in this paper (B-anonymous) obtains better following activity prediction results than the generalised sampling method (G-anonymous). Considering the behavioural constraints, all three metrics get better results than the generalised sampling method (G-anonymous) for the next activity prediction, especially on the Sepsis data, this advantage is undeniable, this is because of the multiple trace variants on the data, which also reflects the defects of the sampling generalisation method when there are many types of traces, sampling generalisation is prone to cause the distribution of data variations. It can amplify the noise contained in the original logs. These results show that the quality of the privacy-processed event logs can be better guaranteed using the proposed method.



a) B-anonymous b) G-anonymous

c) B-anonymous d) G-anonymous



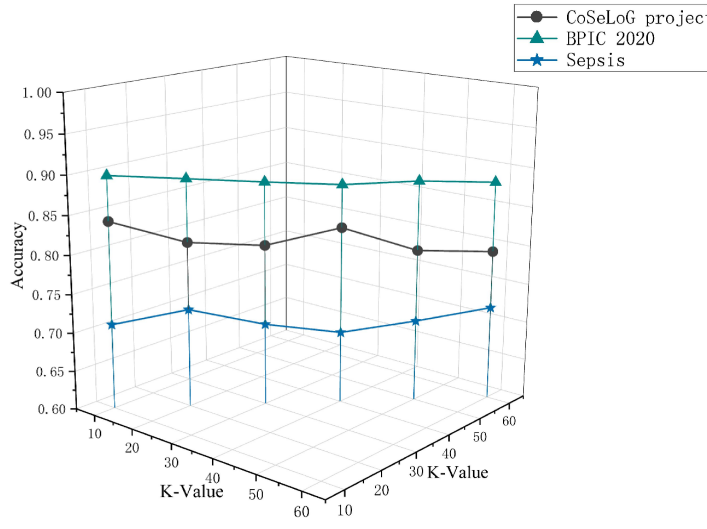
e) B-anonymous f) G-anonymous

Figure 8. Results of different metrics

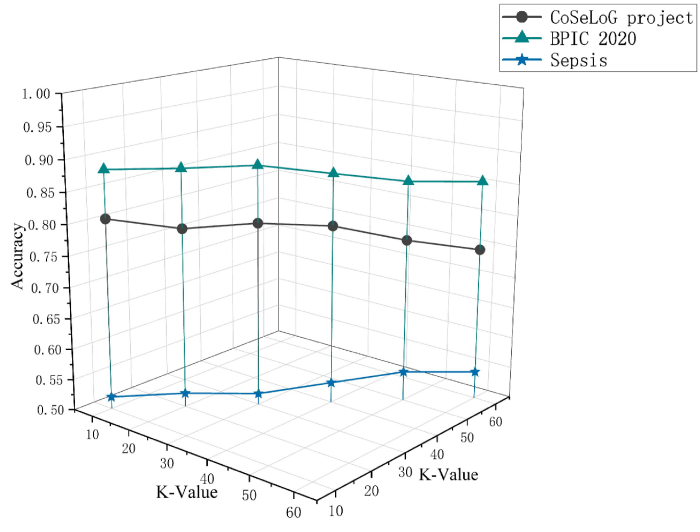
Figure 9 shows the prediction accuracy values obtained by the two methods on different datasets, and it can be observed that the proposed method still achieves better results under this metric. The evaluation of the experiments leads to the conclusion that the method proposed in this paper can achieve behavioural anonymization of event logs while safeguarding the utility of privacy-processed event logs as much as possible.

6 CONCLUSION

Event logs are the starting point for process analysis and optimization, which contain the privacy information of process performers. Traditional approaches protect users' data privacy through generalization and noise insertion, but ignore the privacy leakage problem caused by behavioural relationships between activities. To solve this problem, this paper proposes a method for business process behaviour anonymization based on log skeleton, which firstly models the individual traces of executors in the event log and performs similarity clustering, filters incomplete clusters according to whether the number of behavioural models in the clusters reaches the privacy parameter, and then extracts behavioural relationships in the incomplete clusters according to the behavioural relationships of the log skeleton, constructs the log-skeleton based behavioural constraint set, then extract the possible prefixes in the overall event logs, perform privacy trace generation and privacy model filling



a) B-anonymous



b) G-anonymous

Figure 9. Prediction accuracy of both methods

based on the previously constructed behavioural constraint set to induce the overall behaviour to achieve K-anonymity, and finally conduct experiments on four different event logs. The results show that the method proposed in this paper outperforms the comparison method on various datasets.

The proposed method in this paper focuses on fine-grained behaviours in business processes and is able to deal with possible behavioural privacy attacks in healthcare, finance, and other domains. One future research direction is to consider the privacy release problem under the fusion of behaviour and data, and another direction of interest is the problem of event logging behaviour and data loss metrics after privacy processing, in order to realize the information loss after business process privacy processing in multiple perspectives and dimensions.

Acknowledgements

Supported by the National Natural Science Foundation, China (Nos. 61572035, 61402011), Anhui Provincial Natural Science Foundation (Water Science Joint Fund, No. 2308085US11), Key Research and Development Program of Anhui Province (No. 2022a05020005), the Leading Backbone Talent Project in Anhui Province, China (No. 2020-1-12).

REFERENCES

[1] Beerepoot, I.—Di Ciccio, C.—Reijers, H. A.—Rinderle-Ma, S.—Bandara, W. et al.: The Biggest Business Process Management Problems to Solve Before We Die. Computers in Industry, Vol. 146, 2023, Art. No. 103837, doi: 10.1016/j.compind.2022.103837.

[2] Mannhardt, F.—Petersen, S. A.—Oliveira, M. F.: Privacy Challenges for Process Mining in Human-Centered Industrial Environments. 2018 14th International Conference on Intelligent Environments (IE), IEEE, 2018, pp. 64–71, doi: 10.1109/IE.2018.00017.

[3] Elkoumy, G.—Fahrenkrog-Petersen, S. A.—Sani, M. F.—Koschmider, A.—Mannhardt, F.—Von Voigt, S. N.—Rafiei, M.—Waldthausen, L. V.: Privacy and Confidentiality in Process Mining: Threats and Research Challenges. ACM Transactions on Management Information System (TMIS), Vol. 13, 2021, No. 1, Art. No. 11, doi: 10.1145/3468877.

[4] Elkoumy, G.—Pankova, A.—Dumas, M.: Differentially Private Release of Event Logs for Process Mining. Information Systems, Vol. 115, 2023, Art. No. 102161, doi: 10.1016/j.is.2022.102161.

[5] Elkoumy, G.—Dumas, M.: Libra: High-Utility Anonymization of Event Logs for Process Mining via Subsampling. 2022 4th International Conference on Process Mining (ICPM), IEEE, 2022, pp. 144–151, doi: 10.1109/ICPM57379.2022.9980619.

[6] Batista, E.—Solanas, A.: A Uniformization-Based Approach to Preserve Individuals' Privacy During Process Mining Analyses. Peer-to-Peer Networking and Applications, Vol. 14, 2021, No. 3, pp. 1500–1519, doi: 10.1007/s12083-020-01059-1.

[7] Rafiei, M.—van der Aalst, W. M. P.: Privacy-Preserving Data Publishing in Process Mining. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (Eds.): Business Process Management Forum (BPM 2020). Springer, Cham, Lecture Notes in Business Information Processing, Vol. 392, 2020, pp. 122–138, doi: 10.1007/978-3-030-58638-6_8.

[8] Rafiei, M.—Wagner, M.—van der Aalst, W. M. P.: TLKC-Privacy Model for Process Mining. In: Dalpiaz, F., Zdravkovic, J., Loucopoulos, P. (Eds.): Research Challenges in Information Science (RCIS 2020). Springer, Cham, Lecture Notes in Business Information Processing, Vol. 385, 2020, pp. 398–416, doi: 10.1007/978-3-030-50316-1_24.

[9] Pika, A.—Wynn, M. T.—Budiono, S.—ter Hofstede, A. H. M.—van der Aalst, W. M. P.—Reijers, H. A.: Privacy-Preserving Process Mining in Healthcare. International Journal of Environmental Research and Public Health, Vol. 17, 2020, No. 5, Art. No. 1612, doi: 10.3390/ijerph17051612.

[10] Mannhardt, F.—Koschmider, A.—Baracaldo, N.—Weidlich, M.—Michael, J.: Privacy-Preserving Process Mining: Differential Privacy for Event Logs. Business and Information Systems Engineering, Vol. 61, 2019, No. 5, pp. 595–614, doi: 10.1007/s12599-019-00613-3.

[11] Liu, C.—Duan, H.—Zeng, Q.—Zhou, M.—Lu, F.—Cheng, J.: Towards Comprehensive Support for Privacy Preservation Cross-Organization Business Process Mining. IEEE Transactions on Services Computing, Vol. 12, 2019, No. 4, pp. 639–653, doi: 10.1109/TSC.2016.2617331.

[12] Elkoumy, G.—Fahrenkrog-Petersen, S. A.—Dumas, M.—Laud, P.—Pankova, A.—Weidlich, M.: Secure Multi-Party Computation for Inter-Organizational Process Mining. In: Nurcan, S., Reinhartz-Berger, I., Soffer, P., Zdravkovic, J. (Eds.): Enterprise, Business-Process and Information Systems Modeling (BPMDS 2020, EMMSAD 2020). Springer, Cham, Lecture Notes in Business Information Processing, Vol. 387, 2020, pp. 166–181, doi: 10.1007/978-3-030-49418-6_11.

[13] Cheng, L.—Liu, C.—Zeng, Q.: Optimal Alignments Between Large Event Logs and Process Models over Distributed Systems: An Approach Based on Petri Nets. Information Sciences, Vol. 619, 2023, pp. 406–420, doi: 10.1016/j.ins.2022.11.052.

[14] Fahrenkrog-Petersen, S. A.—Kabierski, M.—van der Aa, H.—Weidlich, M.: Semantics-Aware Mechanisms for Control-Flow Anonymization in Process Mining. Information Systems, Vol. 114, 2023, Art. No. 102169, doi: 10.1016/j.is.2023.102169.

[15] Batista, E.—Martínez-Ballesté, A.—Solanas, A.: Privacy-Preserving Process Mining: A Microaggregation-Based Approach. Journal of Information Security and Applications, Vol. 68, 2022, Art. No. 103235, doi: 10.1016/j.jisa.2022.103235.

[16] Rösel, F.—Fahrenkog-Petersen, S. A.—van der Aa, H.—Weidlich, M.: A Distance Measure for Privacy-Preserving Process Mining Based on Feature Learning. In: Marrella, A., Weber, B. (Eds.): Business Process Management Workshops

(BPM 2021). Springer, Cham, Lecture Notes in Business Information Processing, Vol. 436, 2021, pp. 73–85, doi: 10.1007/978-3-030-94343-1_6.

[17] Agostinelli, S.—Chiariello, F.—Maggi, F. M.—Marrella, A.—Patrizi, F.: Process Mining Meets Model Learning: Discovering Deterministic Finite State Automata from Event Logs for Business Process Analysis. Information Systems, Vol. 114, 2023, Art. No. 102180, doi: 10.1016/j.is.2023.102180.

[18] Bayomie, D.—Di Ciccio, C.—Mendling, J.: Event-Case Correlation for Process Mining Using Probabilistic Optimization. Information Systems, Vol. 114, 2023, Art. No. 102167, doi: 10.1016/j.is.2023.102167.

[19] Pasquadibisceglie, V.—Appice, A.—Castellano, G.—Malerba, D.: DARWIN: An Online Deep Learning Approach to Handle Concept Drifts in Predictive Process Monitoring. Engineering Applications of Artificial Intelligence, Vol. 123, Part C, 2023, Art. No. 106461, doi: 10.1016/j.engappai.2023.106461.

[20] Khan, R.—Tao, X.—Anjum, A.—Kanwal, T.—Malik, S. U. R.—Khan, A.—Rehman, W. U.—Maple, C.: θ-Sensitive k-Anonymity: An Anonymization Model for IoT Based Electronic Health Records. Electronics, Vol. 9, 2020, No. 5, Art. No. 716, doi: 10.3390/electronics9050716.

[21] Verbeek, H. M. W.—de Carvalho, R. M.: Log Skeletons: A Classification Approach to Process Discovery. CoRR, 2018, doi: 10.48550/arXiv.1806.08247.

[22] Galanti, R.—de Leoni, M.—Monaro, M.—Navarin, N.—Marazzi, A.—Di Stasi, B.—Maldera, S.: An Explainable Decision Support System for Predictive Process Analytics. Engineering Applications of Artificial Intelligence, Vol. 120, 2023, Art. No. 105904, doi: 10.1016/j.engappai.2023.105904.

[23] Kratsch, W.—Manderscheid, J.—Röglinger, M.—Seyfried, J.: Machine Learning in Business Process Monitoring: A Comparison of Deep Learning and Classical Approaches Used for Outcome Prediction. Business and Information Systems Engineering, Vol. 63, 2021, No. 3, pp. 261–276, doi: 10.1007/s12599-020-00645-0.

[24] Wang, J.—Gao, S.—Tang, Z.—Tan, D.—Cao, B.—Fan, J.: A Context-Aware Recommendation System for Improving Manufacturing Process Modeling. Journal of Intelligent Manufacturing, Vol. 34, 2023, No. 3, pp. 1347–1368, doi: 10.1007/s10845-021-01854-4.

[25] Fahrenkrog-Petersen, S. A.—van der Aa, H.—Weidlich, M.: PRIPEL: Privacy-Preserving Event Log Publishing Including Contextual Information. In: Fahland, D., Ghidini, C., Becker, J., Marlon, D. (Eds.): Business Process Management (BPM 2020). Springer, Cham, Lecture Notes in Computer Science, Vol. 12168, 2020, pp. 111–128, doi: 10.1007/978-3-030-58666-9_7.

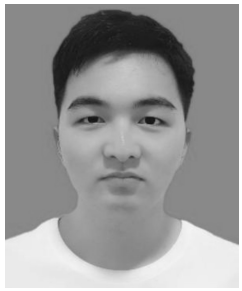
[26] Buijs, J.: Receipt Phase of an Environmental Permit Application Process ('WABO'), CoSeLoG Project. Eindhoven University of Technology, 2014, doi: 10.4121/UUID:A07386A5-7BE3-4367-9535-70BC9E77DBE6.

[27] van Dongen, B.: BPI Challenge 2020: International Declarations. 4TU.Centre for Research Data, 2020, doi: 10.4121/UUID:2BBF8F6A-FC50-48EB-AA9E-C4EA5EF7E8C5.

[28] Mannhardt, F.: Sepsis Cases – Event Log. Eindhoven University of Technology, 2016, doi: 10.4121/UUID:915D2BFB-7E84-49AD-A286-DC35F063A460.

[29] Deng, D.: DBSCAN Clustering Algorithm Based on Density. 2020 7th International Forum on Electrical Engineering and Automation (IFEEA), IEEE, 2020, pp. 949–953, doi: 10.1109/IFEEA51475.2020.00199.

[30] Gao, Y.—Fang, Z.—Xu, J.—Gong, S.—Shen, C.—Chen, L.: An Efficient and Distributed Framework for Real-Time Trajectory Stream Clustering. IEEE Transactions on Knowledge and Data Engineering, Vol. 36, 2024, No. 5, pp. 1857–1873, doi: 10.1109/TKDE.2023.3312319.



Xinsheng Fang is currently pursuing his Ph.D. in information security engineering at the Anhui University of Science and Technology in China. His research interests include process mining, anomaly detection, and privacy protection.

Xianwen Fang received his Ph.D. in computer software and theory from the Tongji University in 2011. Currently, he is Professor of information security engineering at the Anhui University of Science and Technology. His research interests include Petri nets, trusted software and big data.

Ke Lu received his Ph.D. in information security engineering from the Anhui University of Science and Technology in China. Currently, he is a Lecturer in information security engineering at the Anhui University of Science and Technology. His research interests include deep learning, process predictive monitoring, and process mining.