



# Applied Artificial Intelligence

## An International Journal

ISSN: (Print) (Online) Journal homepage: <https://www.tandfonline.com/loi/uaai20>

## Event Log Privacy Based on Differential Petri Nets

Daoyu Kan, Xianwen Fang & Ziyong Gong

To cite this article: Daoyu Kan, Xianwen Fang & Ziyong Gong (2023) Event Log Privacy Based on Differential Petri Nets, Applied Artificial Intelligence, 37:1, 2175109, DOI: [10.1080/08839514.2023.2175109](https://doi.org/10.1080/08839514.2023.2175109)

To link to this article: <https://doi.org/10.1080/08839514.2023.2175109>



© 2023 The Author(s). Published with license by Taylor & Francis Group, LLC.



Published online: 23 Feb 2023.



Submit your article to this journal [↗](#)



Article views: 510




View related articles [↗](#)



View Crossmark data [↗](#)

## Event Log Privacy Based on Differential Petri Nets

Daoyu Kan <sup>a</sup>, Xianwen Fang <sup>a,b</sup>, and Ziyou Gong <sup>a</sup>

<sup>a</sup>School of Mathematics and Big Data, Anhui University of Science and Technology, Huainan, China;

<sup>b</sup>Anhui Province Engineering Laboratory for Big Data Analysis and Early Warning Technology of Coal Mine Safety, Huainan, China

### ABSTRACT

Process mining uses event logs to improve business processes, but such logs may contain privacy information. One popular research problem is the privacy protection of event logs. Publishing logs with differential privacy is one of major research directions. Existing research achieves privacy protection primarily by injecting random noise into event logs, or merging similar information. The former ignores the fact that injecting random noise will produce apparently unreasonable activity traces, and the latter will cause a loss of process information in the process mining perspective. To solve the above problems, this article proposes a differential algorithm based on randomized response to model Petri nets for the original event logs, select the important labels in the logs by the weak sequential relationship of control flow between activities, inject noise into the Petri net model based on the important labels using the randomized response approach, and establish a differential Petri net model. Experiments on public datasets show that the event logs produced by the approach proposed do not contain unreasonable traces. Compared with the baseline approach, the proposed approach performs better on Fitness metrics with consistent privacy requirements and retains more process variants, reducing the loss of original event log process information.

### ARTICLE HISTORY

Received 12 December 2022

Revised 14 January 2023

Accepted 27 January 2023

## Introduction

Process mining aims to help organizations or individuals to improve the performance, conformance or quality of their relevant business processes. The event logs recorded by information systems are the starting point of process mining (van der Aalst 2016), which relies on the event logs of business processes (van der Aalst 2012) to obtain information from the event logs to discover, monitor or improve the actual business processes. The process discovery algorithm constructs a process model from the event log in order to define the relationship between activities in the process (Augusto et al. 2019).

Event logs contain information about individuals with corresponding cases, and such information may contain private information directly, or private information about specific individuals may be obtained from such

**CONTACT** Xianwen Fang  [xwfang@aust.edu.cn](mailto:xwfang@aust.edu.cn)  School of Mathematics and Big Data, Anhui University of Science and Technology, 168 Taifeng Street, Huainan, Anhui 232001, China

© 2023 The Author(s). Published with license by Taylor & Francis Group, LLC.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

information, so that event logs are potentially invasive of personal privacy (Mannhardt, Petersen, and Oliveira 2018). For example, for Healthcare Process, an event log may include personal information about the patient receiving the treatment, such as (Stefanini et al. 2018) and (Liu et al. 2022). (Nuñez von Voigt et al. 2020) used individual uniqueness in the event log to quantify the risk of re-identification, and showed that all cases in the event log are likely to be re-identified. In order to protect the private information of individuals in event logs, most countries and regions have restricted the analysis of such event logs by introducing regulations, such as the European General Data Protection Regulation (GDPR), which prohibits the processing of personal data except for specific needs (Voss 2016).

In order to circumvent legal restrictions, (Elkoumy et al. 2021) proposed privacy-preserving mining that aims to protect personal data while being able to improve business processes. Existing research has focused on implementing privacy-preserving process mining from two different perspectives (Fahrenkrog-Petersen 2019): anonymizing the event logs used for process mining (Fahrenkrog-Petersen, van der Aa, and Weidlich 2020), or designing process mining algorithms so that their outputs satisfy privacy requirements (Kabierski, Fahrenkrog-Petersen, and Weidlich 2021). In this article, we focus on the perspective of anonymizing event logs, and existing research has proposed numerous approaches to anonymize event logs. (Rafiei, von Waldthausen, and van der Aalst 2020) proposed the definition of distance metric for traces through embedding of events based on distance metric of feature learning and guided the anonymization of event logs. (Pika et al. 2020) evaluated the applicability of privacy-preserving data transformation techniques in the field of data mining for the anonymization of medical process data in the context of medical processes and demonstrated how some of the methods affect the process mining results.

However, anonymization of data only does not effectively protect against differential attacks. For example, in the case of healthcare logs, if an external attacker has prior knowledge that only a particular patient has performed a particular activity, then anonymizing only the relevant information in the data, such as the patient's name, will not achieve the privacy protection goal. To prevent such attacks, (Dwork 2008) proposed the concept of differential privacy. (Mannhardt et al. 2019) investigated potential privacy breaches and means of protection from common assumptions about event logs used in process mining. A model for the protection of event data privacy is developed. (Elkoumy, Pankova, and Dumas 2021) proposed a differential privacy mechanism that oversamples the cases in the logs and adds noise to the timestamps to ensure that the probability of an attacker identifying any individual in the original logs does not increase beyond a threshold after the anonymized logs are made public. (Rösel et al. 2022) incorporated a distance metric based on feature learning to consider activity semantics in the

anonymization process. (Fahrenkrog-Petersen, van der Aa, and Weidlich 2019) and (Rafiei, Wagner, and van der Aalst 2020) combine distance metrics to merge multiple different traces of a process to achieve hiding information about a single process instance, as well as the personal information of the person associated with that instance. (Elkoumy et al. 2022) proposed a tool for inter-organizational process mining privacy protection that enables independent parties to perform process mining operations without revealing any data. (Feng et al. 2018) proposed a differential privacy collaborative filtering recommendation algorithm based on behavioral similarity. By adding noise obeying Laplace distribution to the behavioral similarity matrix, the privacy of the correlator is effectively protected. (Batista and Solanas 2021) proposed an approach based on uniformization of event distribution to protect personal privacy in process mining. (Hou et al. 2022) introduced fuzziness into differential privacy to establish fuzzy differential privacy to achieve a more flexible balance between the accuracy of the output and the level of privacy protection of the data.

Privacy protection of personal information in event logs through differential privacy has been a popular direction in recent years, and many researchers have proposed well-established solutions. However, there are several problems with existing research that may have a negative impact on the application of process mining techniques. One of such problems is the introduction of random noise into the event logs, which may produce traces that are not present in the original event logs, or traces with obvious logical errors. For example, in a healthcare process, assuming that in a certain anonymized trace, the patient first performs the activity of “pharmacy medicine pickup” and then the activity of “outpatient consultation,” this is clearly not logical and makes the attacker doubt the authenticity of the trace and choose not to trust this activity trace, reducing the privacy protection performance. The second is to prohibit the publication of infrequent activity traces in the event log, represented by the k-anonymity approach (Sweeney 2002), or to merge similar information in the event log, as in (Fahrenkrog-Petersen, van der Aa, and Weidlich 2019). From the perspective of process mining, the disadvantage of these approaches is that part of the process information is lost, and the information contained in the original logs is hidden. This obviously has a negative impact on the effectiveness of process mining techniques. The research in this article is based on the above two problems and is summarized in two research questions.

**RQ1:** How to avoid the differential mechanism that generates event logs to produce unreasonable activity traces?

**RQ2:** How to reduce the loss of process information while ensuring the degree of privacy of the event log?

The goal that this article hopes to achieve is to reduce the loss of the original process information caused by the processing of the event logs, while ensuring the privacy of the processed event logs. In addition, ensure that the resulting new event log does not contain traces that are not present in the original log. Based on this goal, we propose a randomized response-based differential algorithm, an approach that solves the two research problems we have proposed by publishing a differential Petri net model that balances privacy protection performance with data availability while ensuring data security, and validates it through experiments. This approach models Petri nets on the original data by process mining approach, obtains important label sets based on the weak sequential relationships of control flow between activities, builds randomized anonymization structures, and generates differential Petri net models. For RQ1, this article selects the activity labels added as noise based on the weak order relationship of control flow, and the produced traces are all included in the original event log, and will not produce traces that do not exist in the original event log or apparently do not conform to the common sense of life. For RQ2, this article builds a differential Petri net model based on the randomized response approach with the weak sequential relationship between activities to generate a privacy event log and reduce the loss of process information.

## Related Knowledge

In this section, we describe the basic concepts involved in this article. [section 2.1](#) introduces the basic concepts of Petri nets and behavioral profile. [section 2.2](#) introduces the concept of differential privacy and its related definitions.

### Petri Net

**Definition 1 (Petri Net)** Let  $N = (P, T; F)$  be a net, Mapping  $M: P \rightarrow \{0, 1, 2, \dots\}$  is called the mark of the net  $N$ , and the quaternion  $(P, T; F, M)$  is called the marked net i.e. Petri net.

There exists a weak order relation in Petri net PN, i.e., the sequence  $\delta = t_1 t_2 \dots t_n$  for the arbitrary activity pair  $(x, y)$  containing  $T \times T$  when  $i \in \{1, 2, \dots, n-1\}$ ,  $i < j \leq n$   $t_i = x$  and  $t_j = y$ ,  $x \succ y$ .

- (1) Strict Order Relation  $\rightarrow$ , If and only if  $x \succ y, y \not\succ x$ .
- (2) Exclusiveness Relation  $+$ , If and only if  $x \not\succ y, y \not\succ x$ .
- (3) Interleaving Order Relation  $||$ , If and only if  $x \succ y, y \succ x$ .

**Definition 2 (Behavioral profile)** (Weidlich et al. 2010) Let  $PN = (P, T; F)$  be a Petri net, For any transition pair  $(x, y) \in (T \times T)$  satisfying one of the following relations.

### **Differential Privacy**

Differential privacy is a new definition of privacy proposed for the privacy leakage problem of databases. Mainly by using random noise to ensure that, the result of query requesting publicly visible information does not reveal individual privacy information, and individual characteristics are removed to protect user privacy while preserving statistical characteristics, so to some extent, differential privacy can ensure the availability of the protected data so that it can still be applied in the field of data analysis. Nevertheless, the usability of the data is reduced.

As a real-life example, in the mid-1990s, the state of Massachusetts released anonymized employee medical records for research purposes, and an MIT graduate student was able to decipher the anonymized medical records by matching them with publicly available voter registration records to obtain information about the governor. The purpose of proposing differential privacy is to avoid such breaches of data privacy as much as possible and to reduce data leakage when the data is used for research.

Differential privacy hopes to achieve the goal that, while the publicly available dataset is analyzed for valid data information about groups in the dataset, there is no increase in the knowledge of information about specific individuals in the dataset, and the presence or absence of information about any individual in the dataset does not affect the results of differential privacy. That is, the purpose of differential privacy is to reduce the impact of individual data on the overall query results.

**Definition 3 (Neighboring datasets)** Given two different data sets  $D_1$  and  $D_2$ ,  $D_1$  and  $D_2$  are called neighboring datasets when  $D_1$  and  $D_2$  satisfy that there is and only one data difference.

According to Definition 3, dataset (1) and dataset (2) and dataset (1) and dataset (3) in [Figure 1](#) all constitute neighboring datasets, since they all differ from each other by only one piece of data.

**Definition 4 (Differential privacy)**  $M$  is a randomized algorithm. The datasets  $D_1$  and  $D_2$  are neighboring datasets.  $E$  is an arbitrary subset of all possible outputs of the randomized algorithm  $M$ . If there is  $Pr[M(D_1) \in E] \leq exp(\epsilon) \times Pr[M(D_2) \in E]$ . Then the algorithm  $M$  is said to be  $\epsilon$ -differential privacy.

CaseID	Gender	Married?
1	F	YES
2	M	YES
3	F	NO
4	M	NO

(1)

CaseID	Gender	Married?
1	F	YES
2	M	NO
3	F	NO
4	M	NO

(2)

CaseID	Gender	Married?
1	F	YES
2	M	YES
3	F	NO
4	M	NO
5	M	YES

(3)

**Figure 1.** Neighboring datasets.

$\epsilon$  is the privacy budget that controls the degree of privacy protection of the algorithm. the smaller the  $\epsilon$ , the higher the degree of privacy protection.

**Definition 5 (Post-processing immunity)**  $MN^{|x|} \rightarrow R$  is a randomized algorithm for differential privacy satisfying  $\epsilon$ -differential privacy, and  $f : R \rightarrow R'$  is an arbitrary randomized mapping, then  $f \circ M : N^{|x|} \rightarrow R'$  is  $\epsilon$ -differentially private.

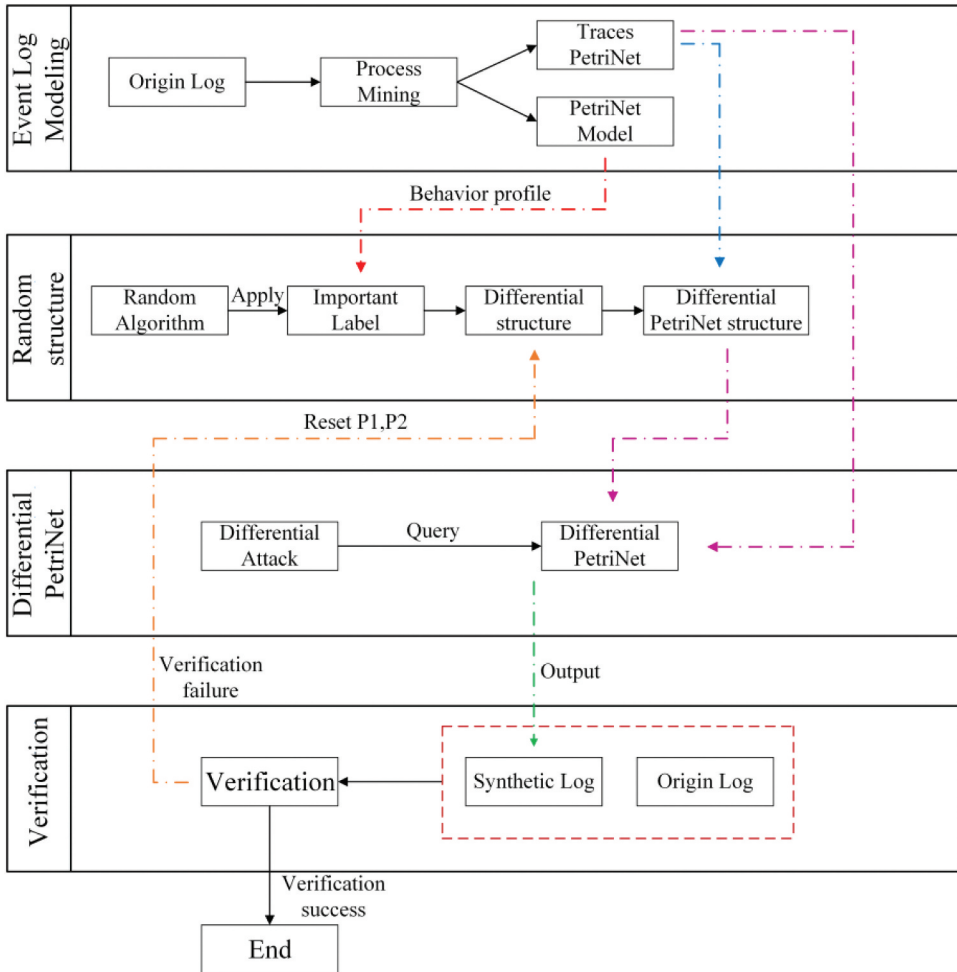
The meaning of Definition 5 is that the differential privacy algorithm is independent of post-processing, and the result obtained by combining the differential privacy algorithm with an arbitrary mapping is still consistent with differential privacy, also known as differential privacy post-processability.

## Differential Petri Nets

To address the two research questions presented in the first section, this article proposes a differential algorithm based on randomized response that mines the Petri net model of the original event log, selects important labels by weak sequential relationships among activities, adds these labels to the Petri net model as noise, builds the differential Petri net model and publishes it instead of the event log. Since the important labels are present in the original event log, the algorithm does not generate traces or behaviors that are not present in the original event log. The model responds to queries by synthesizing a virtual event log, which is  $\epsilon$ -differentially private according to the post-processing nature of differential privacy. At the same time, the algorithm still keeps the virtual event log data with certain availability. In this section, the algorithm is illustrated in this article with a simple simulated event log as an example. The steps of the algorithm are shown in [Figure 2](#).

## Event Log

[Table 1](#) shows the event log of a hospital diagnostic process. Each different kind of event trace corresponds to a different diagnostic process. In this article, we use a process mining algorithm to model this event log as a Petri net and



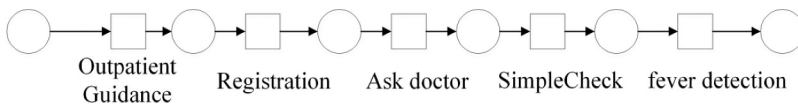
**Figure 2.** Algorithm process.

**Table 1.** Event log.

	Variant	#
$s_1$	Outpatient Guidance,Registration,Ask doctor,SimpleCheck, fever detection	40
$s_2$	Outpatient Guidance,Registration,Ask doctor,SimpleCheck, AIDS detection	5
$s_3$	Outpatient Guidance,Ask doctor, Registration,SimpleCheck, fever detection	25
$s_4$	Outpatient Guidance,Ask doctor, Registration,SimpleCheck, AIDS detection	10
$s_5$	Outpatient Guidance,Registration,Ask doctor, Change doctor,SimpleCheck, fever detection	3

obtain the Petri net model corresponding to the original event log. The process mining approach is a process-oriented modeling and analysis method, the idea of which is to obtain information from event logs to discover, improve or monitor actual business processes. In the past decades, many kinds of process mining algorithms have been proposed, such as  $\alpha$ -algorithm (van der Aalst, Weijters, and Maruster 2004), Inductive algorithm (Leemans, Fahland, and van der Aalst 2013), HPNs (Liu et al. 2022), MBPM (Liu 2022), etc. In this





**Figure 3.** Model of trace.

section, we use Inductive algorithm to model the event logs and apply process mining algorithms to each different kind of process variants in the event logs to obtain the corresponding Petri net models. Figure 3 shows the model for one of the classes of traces.

### **Private Label**

Business process management has grown in importance over the past few decades. However, event logs often contain users' personal private information, so many countries and regions restrict data analysis of these event logs. In the hospital diagnosis process presented in this article, the event labels may contain private information about the patient. For example, in the event log shown in Table 1, for the activity AIDS detection; whether this patient is eventually diagnosed with AIDS or not, he does not want this information to be disclosed. However, by querying the public event log, it is theoretically possible to know whether a patient has performed a specific activity.

Assuming that Patient A is known to be in a public event log, combine the results of the following two queries "What is the number of people with active AIDS detection in this event log?" and "What is the number of people in this event log who have active AIDS detection other than Patient A?" This type of method of obtaining privacy information through several queries is called differential attack, and in order to defend against this type of attack, it is necessary to introduce some inaccuracy into the event log, i.e., to introduce noise into the event log.

There have been studies on differential privacy protection of event logs by randomly injecting noise, and such methods may produce traces that are not present in the original logs themselves or are obviously not in line with common sense, reducing the effectiveness of differential privacy. The algorithm proposed in this article performs differential privacy protection for event logs by using differential Petri nets instead of traditional event log publication, and in response to queries, differential Petri nets generate virtual event logs that are also differentially private.

Applying noise to event logs leads to a reduction in the availability of event log data, which is required by process analysis algorithms, so we want to be able to protect personal private information while keeping the event log data still available. If all activity tags in the trace are anonymized, the usability of the event log data will be significantly reduced. In fact, not all activity tags contain

privacy information; for example, in healthcare scenarios, different medical conditions may have common activity labels, such as blood draw, temperature measurement, etc. Only a few labels, such as AIDS detection, will contain privacy information. Therefore, to ensure data availability, in this article, only activity labels that may contain privacy information are anonymized. We refer to the labels of such activities that contain privacy information as private labels, and in the work done in this paper, the private labels are obtained from a priori knowledge, such as AIDS detection, a private activity in process, which is based on the experience in daily life.

**Definition 6 (Sensitive labels)**  $N = (P, T; F, M)$  is a Petri net,  $T = (t_1, t_2, \dots, t_n)$  is the transition set of this Petri net, the label of  $t_i \in T$  is a private label, and if  $t_j \in T$  and  $t_i + t_j$ , the label of  $t_j$  is a sensitive label in Petri net  $N$ .

As an example, the event log in Table 1 is assumed to correspond to a Petri net model in which the privacy label is AIDS detection, and this conclusion is obtained from a priori knowledge. According to the definition of the previous behavioral profile relation and sensitive label, the corresponding sensitive label in this Petri net model can be obtained as fever detection. For the convenience of description, the private label in a Petri net and its corresponding sensitive label are collectively referred to as important labels in this article.

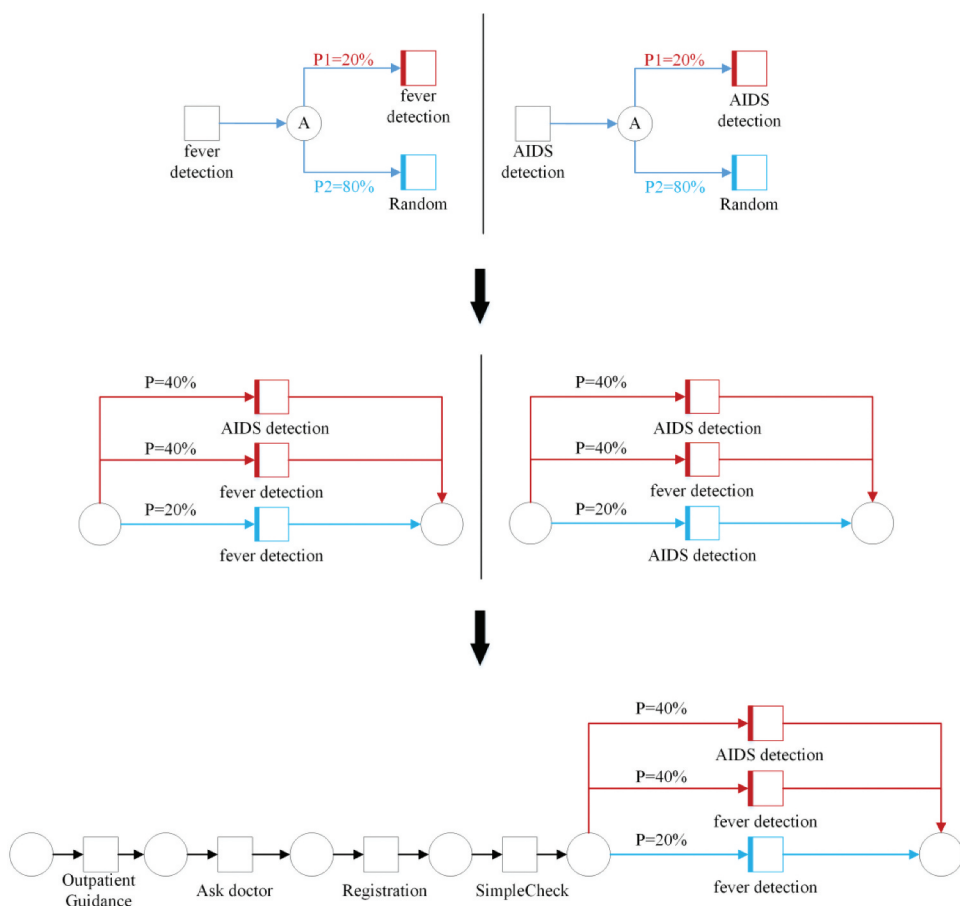
**Definition 7 (Important labels)**  $N = (P, T; F, M)$  is a Petri net,  $T = (t_1, t_2, \dots, t_n)$  is the transition set of this Petri net,  $T_p$  is the privacy label set,  $T_s$  is the sensitive label set. Then  $T_{im} = (T_p, T_s)$  is the set of important labels, and arbitrary  $t_i \in T_{im}$  is a important label.

### Randomization Algorithm

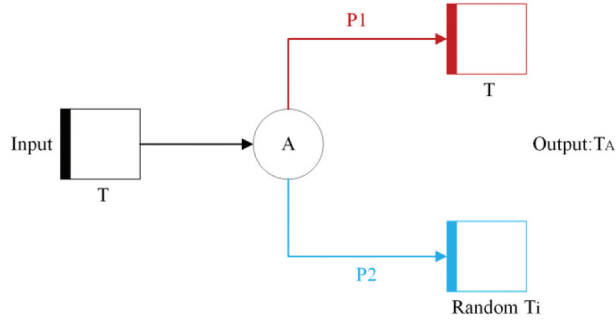
According to the concept of differential privacy, in order to achieve the protection of the private information of the case entity, i.e., the patient, in the event log, it is first necessary to randomize the important labels in this Petri net model to achieve anonymity. In this article, a randomized response algorithm is introduced to randomize the important labels. In life, we often encounter the situation that in a survey, when a question involves the privacy of the respondent, it may happen that the respondent does not want to give the true answer, and a solution is to let each respondent add noise to their respective answers. For example, suppose the interviewer asks a sensitive question about right and wrong, the respondent can flip a coin once and

answer the true answer if the result of the coin flip is heads, or randomly answer yes or no if it is tails, which is the concept of randomized response. The process of obtaining the differential Petri net corresponding to a case in the event log using the randomized response algorithm is shown in [Figure 4](#).

In this article, we introduce the concept of randomized response into the Petri net model to anonymize the important labels in the model. In Petri nets, we cannot reduce the variation in Petri nets to simple right and wrong questions, and need to make optimization of randomized responses. In the traditional randomized responses algorithm, the algorithm outputs the true value with a certain probability, while the remaining cases are randomly outputted as yes or no answers to achieve privacy protection. Mapping to Petri nets, we specify that for important labels in the model, the randomization algorithm outputs their true labels with a certain probability, while the remaining situations randomly output arbitrary important labels with equal probability of outputting different arbitrary labels, and maintain the as-is output for traces that do not contain private labels. The structure of the optimized randomization algorithm is shown in [Figure 5](#).



**Figure 4.** Differential Petri nets obtained by randomizing responses.



**Figure 5.** Randomization algorithm for Petri net labels.

In [Figure 5](#),  $T_i$  is the set of private and sensitive labels in the Petri net model, i.e., the set of important labels.  $T \in T_i$  is the input of the algorithm,  $P1$  is the true output probability, which represents the probability of outputting the true activity labels, i.e., “occasions when a coin is tossed heads,” and  $P2$  is the noise addition probability, which represents the probability of adding noise to the output activity labels, i.e., “occasions when a coin is tossed tails.” In such scenarios, the algorithm randomly selects the activity label in  $T_i$  as the output. It should be noted that the randomization algorithm may still output the true activity label even in the case of “tails of a coin.” After the randomization algorithm, the output label is said to be randomized anonymization. The randomization algorithm is shown in [Algorithm 1](#).

### Algorithm 1: Randomized anonymization

---

**Input:**  $L_1$   
**Output:**  $T_A$

```

1  $p = \text{CaseNumber}(L_1)$  //Number of cases in the event log
2  $L_1 = [C_i | i \in (1, p)]$  //L consists of p cases
3  $M = \text{ProcessMine}(L_1)$  //Mine petri model of  $L_1$ 
4  $m_{total} = [m_i | i \in (1, p)]$  //Initialize case petri model set
5 for  $i=1$  to  $p$  do
6    $m_i = \text{ProcessMine}(C_i)$  //Mining petri models for each case
7 end
8  $T_p = \text{PriorKnowledge}(L_1)$  //Get private transition according to prior knowledge
9  $T_s = \text{Exclude}(T_p, M)$  //Get sensitive transition according to model and  $T_p$ 
10  $T_i = \text{Combine}(T_s, T_p)$ 
11 for  $T \in T_s$  do
12   if  $P = P1$  then
13      $T_A = T$  //Output real activity labels;
14   else
15      $T_A = \text{Random}(T_i)$  //Random output important transition
16   end
17 end
18 return  $T_A$ 

```

---

**Definition 8 (Randomized anonymization)**  $N = (P, T; F, M)$  is a Petri net,  $T_{im} = (t_1, t_2, \dots, t_n)$  is the set of important labels in the net  $N$ ; a randomized anonymous  $C$  is a transformation with an input arbitrary transition  $t_i \in T_{im}$  and an output transition  $t_j = C(t_i)$ .

In order to achieve differential privacy protection against differential attacks on patients' personal privacy information in event logs, it is necessary to accurately select the true output probability  $P1$  and the noise addition probability  $P2$ . In the algorithm of this article, firstly, the initial probabilities of  $P1$  and  $P2$  are set; subsequently, according to Definition 4, the differential privacy mechanism should satisfy Equation (1). Finally, this  $P1$  and  $P2$  are verified using Equation (1) to find the exact values of  $P1$  and  $P2$ .

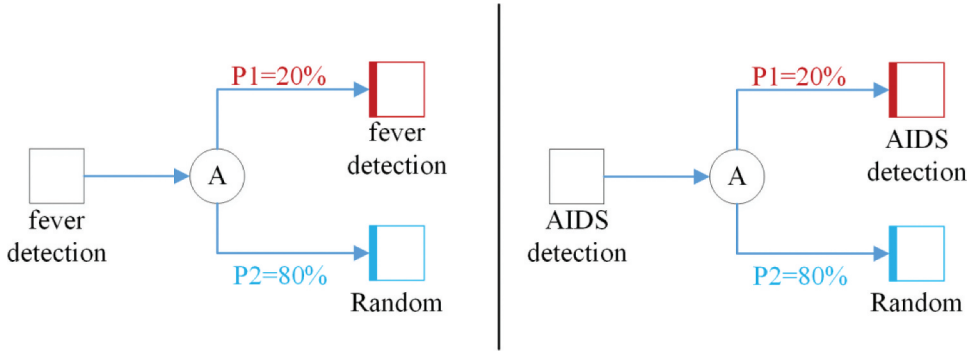
$$Pr[M(D_1) \in S] \leq \exp(\epsilon) \times Pr[M(D_2) \in S] \quad (1)$$

As can be observed from Section 3.2, the Petri net corresponding to the event logs in Table 1 has the private label of AIDS detection and the sensitive label of fever detection, and we take this Petri net model as an example to illustrate the process of randomized anonymization. We set the privacy budget  $\epsilon$  for differential privacy in this example to 1.05, and the initial probabilities of true output probability  $P1$  and noise addition probability  $P2$  are  $P1 = 0.2$  and  $P2 = 0.8$ . According to the Petri net label randomization algorithm structure in Figure 6, we obtain the randomization algorithm structure of the important labels in the example Petri net model, and according to Definition 8, we perform Randomized anonymization. The structure is shown in Figure 6.

### Differential Petri Nets

In section 3.3 the randomized anonymization structure is obtained, which is based on the work done in section 3.2. In this section, the differential Petri net model will be built based on the work done above.

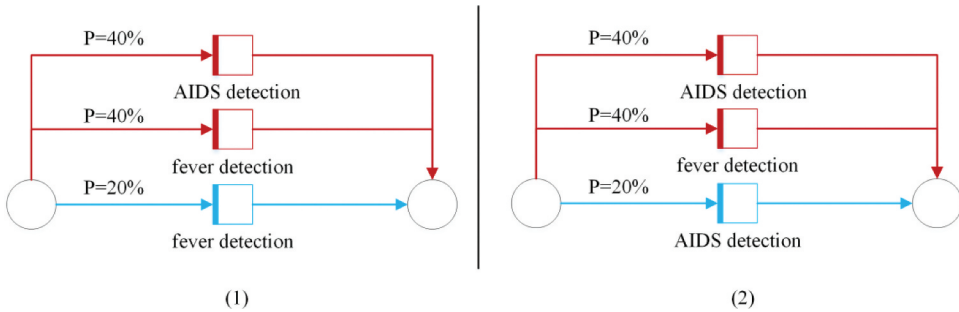
After obtaining the randomized anonymization structure shown in Figure 6, in order to add this structure to the original Petri net, it is necessary to replace this algorithmic structure with the Petri net structure. According to the definition of randomized anonymization, a randomized anonymization transformation  $C$  can be understood as the structure of “place to transition to place” in Petri net, and we use this as the basis for replacing the randomized anonymization structure shown in Figure 6 with the Petri net structure, and the arcs in the Petri net are labeled according to the true output probability  $P1$  and the noise addition probability  $P2$  in Figure 6, and the replacement structure is shown in Figure 7. We use different colors to distinguish the occasions of true output from the occasions of noise addition. In the case of outputting the true activity label, the trace is shown as the blue arc in Figure 7,



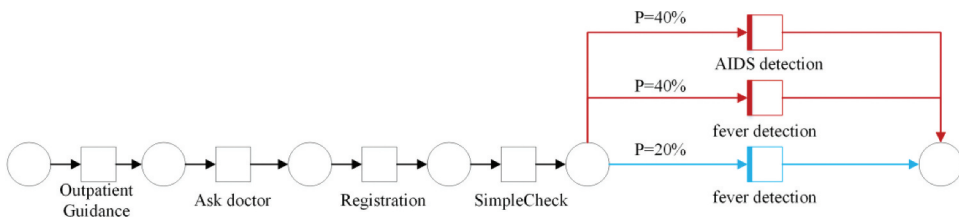
**Figure 6.** Randomized anonymization of important labels.

and when injecting noise into the output activity label, the trace is shown as the red arc in [Figure 7](#).

After obtaining the Petri net structure corresponding to the randomized anonymization, we take the Petri net corresponding to the trace in [Figure 3](#) as an example to obtain the differential Petri net. It is observed that this Petri net model contains sensitive label fever detection, so the structure in [Figure 7\(1\)](#) is used to replace the sensitive label fever detection in the Petri net, and the result is shown in [Figure 8](#), and the replaced Petri net has the structure of randomized anonymity. According to the concept of differential privacy mechanism, the model can synthesize activity traces with differential privacy



**Figure 7.** Randomized anonymization Petri net structure.



**Figure 8.** Differential Petri nets.

protection due to the introduction of noise in the Petri net model, which can achieve differential privacy protection for patient privacy information in the event log, i.e., a differential Petri net model is obtained. Next, we replace the important labels in the Petri nets corresponding to the other activity traces contained in the original logs to obtain the differential Petri nets corresponding to each different activity trace, and these models can generate activity traces with differential privacy, and combine the activity traces corresponding to the differential Petri net model for each trace to obtain the event log with differential privacy protection logs to achieve the protection of patient privacy information in the event logs.

### ***Differential Privacy Protection Event Log and Its Availability Verification***

Section 3.4 obtains the differential Petri net model by introducing the randomized anonymization structure into the Petri net model. According to the concept of differential privacy mechanism, the data generated by this Petri net model are all data with differential privacy, which can achieve differential privacy protection of event logs and avoid the leakage of patient privacy information in event logs by differential attacks. Based on the work in Section 3.4, we replace all the important labels in the Petri net corresponding to each different trace to obtain the differential Petri net model corresponding to each trace, generate the event traces using these models, and combine these activity traces into a new event log that is  $\epsilon$ -differentially private according to the concept of differential privacy.

In this section, the event log of the medical diagnosis process in Table 1 will be used as an example to show how the differential Petri net proposed in this paper can protect the information of patients in the event log with differential privacy. In particular, in addition to preventing differential attacks and leakage of personal privacy information, it is also necessary to ensure that the algorithm-processed data remains somewhat availability. One of the motivations of this article is to reduce the loss of process information in the processed event log, while the frequency information of the process variants in the original log can be restored. If the availability of the event log data is low after processing by the randomization algorithm, the significance of analyzing the event log is lost. Therefore, after applying the randomization algorithm, the availability of the data is also analyzed in this article. The differential privacy log generation algorithm in this article is shown in Algorithm 2.

**Algorithm 2:** Differential privacy log algorithm

---

**Input:**  $T_x, m_{total}, L_1$   
**Output:**  $L_o$

```

1  $p_1 = 0.2$ 
2  $p_2 = 0.8$  //Set initial probability
3  $u = 0.05$  //Set update step
4  $T_i = \text{Loganalysis}(L_1)$  //Important transition set.Details in algorithm 1
5  $q = \text{Number}(T_i)$  //Number of elements in  $T_i$ 
6  $T_i = [t_i | i \in (1, q)]$ 
7  $S = [s_i | i \in (1, q)]$  //Initialize Petri net structure set
8 for  $i=1$  to  $q$  do
9    $t_i = \text{RandomAnonymize}(t_i)$  //Random Anonymize.Details in algorithm 1
10   $s_i = \text{PChange}(t_i)$  //Transform into Petri net structure
11 end
12  $p = \text{CaseNumber}(L_1)$  //Number of cases in the event log
13  $m_{total} = [m_i | i \in (1, p)]$  //Case petri model set
14  $L_{total} = [l_i | i \in (1, p)]$  //Initialize case event log set
15 for  $i=1$  to  $p$  do
16    $m_i = \text{PReplace}(m_i, S)$  //Replace the structure of case Petri net
17    $l_i = \text{GenerateLog}(m_i)$  //Generate virtual event log of case
18 end
19  $L_s = \text{CombineLog}(L_{total})$  //Combine case event log
20 for  $L_p$  do
21   if  $\text{Verify}(L_s) = 1$  then
22      $L_o = L_s$  //Result satisfy Formula 1,output  $L_s$ 
23   else
24      $p_1 = p_1 - u$ 
25      $p_2 = p_2 + u$ 
26     RestartAlgorithm //Result not satisfy Formula 1,Restart algorithm with new
       $p_1, p_2$ 
27   end
28 end
29 return  $L_o$ 

```

---

In [Section 3.4](#), we obtained the differential Petri net model corresponding to each process variant in the original log, and [Figure 8](#) shows the differential Petri net model corresponding to one variant in the original log. According to the post-processing definition of differential privacy, if a mechanism is satisfying differential privacy, the result obtained is consistent with differential privacy no matter what post-processing is applied to it. Therefore, we can synthesize virtual event logs by differential Petri nets, and these synthesized event logs are all differential privacy compliant. Also, since this article uses the activity labels present in the original logs to replace the important labels, no traces or activities that do not exist in the original logs are generated.

The differential Petri nets corresponding to process variants can generate activity traces with differential privacy protection based on probability, and the activity traces generated by differential Petri nets corresponding to all process variants in the event log are recombined to obtain a virtual event log, which is  $\epsilon$ -differentially private according to the post-processing definition of differential privacy. As shown in [Figure 9](#). [Table 2](#) shows a virtual event log synthesized by differential Petri nets, and it should be noted that since there are traces in the original log that do not contain private labels, for such traces



we maintain the output as is and do not process them. According to Definition 5, this event log is  $\varepsilon$ -differentially private.

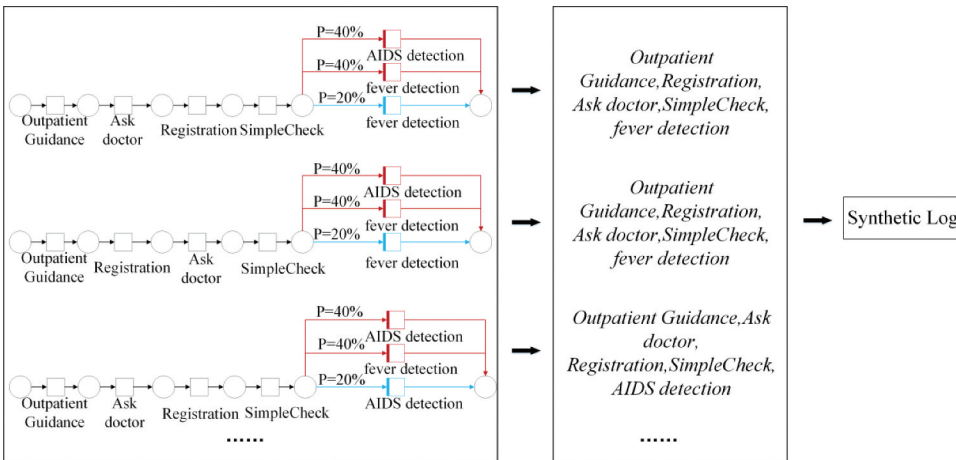
To verify the differential privacy performance of this event log, by definition, the verification formula in this example is shown below.

$$\frac{\Pr(\text{SyntheLabel} = 'AIDSdetection' | \text{Truth} = 'AIDSdetection')}{\Pr(\text{SyntheLabel} = 'AIDSdetection' | \text{Truth} = 'feverdetection')} \quad (2)$$

In this example,  $\text{Truth} = 'AIDSdetection'$  and  $\text{Truth} = 'feverdetection'$  can be regarded as neighboring databases, i.e., the true activity labels of the activities recorded in the event log. This is because in the randomized anonymization structure used in this article, only the label fever detection and the label AIDS detection is performed to replace the original labels, so that no activity traces that do not exist in the original event log or activity traces that are clearly not in line with common sense of life are generated.

$\Pr(\text{SyntheLabel} = 'AIDSdetection' | \text{Truth} = 'AIDSdetection')$  represents the probability that the real label in the original log is AIDS detection, while the activity label in the output virtual event log is AIDS detection, hereafter referred to as Pr1.

$\Pr(\text{SyntheLabel} = 'AIDSdetection' | \text{Truth} = 'feverdetection')$  represents the probability that the real label in the original log is fever detection and the



**Figure 9.** Generate virtual event logs.

**Table 2.** Synthetic event log.

$s_1$	Outpatient Guidance,Registration,Ask doctor,SimpleCheck, fever detection	26
$s_2$	Outpatient Guidance,Registration,Ask doctor,SimpleCheck, AIDS detection	19
$s_3$	Outpatient Guidance,Ask doctor, Registration,SimpleCheck, fever detection	19
$s_4$	Outpatient Guidance,Ask doctor, Registration,SimpleCheck, AIDS detection	16
$s_5$	Outpatient Guidance,Registration,Ask doctor, Change doctor,SimpleCheck, fever detection	3

active label in the output virtual event log is AIDS detection, hereinafter referred to as Pr2.

According to Algorithm 2, the initial probabilities set are  $P1 = 0.2$  and  $P2 = 0.8$ , which are calculated as  $Pr1 = 0.6$  and  $Pr2 = 0.4$ , and the result of the formula is 1.5, i.e.,  $e^{ln1.5}$ , and the privacy budget  $\epsilon$  is  $ln1.5$ , which satisfies  $ln1.5$  differential privacy; the actual privacy requirements will limit the value of the privacy budget to ensure the effect of privacy protection. In this example we set the value of the privacy budget to 1.05, and  $ln1.5 < 1.05$ . According to Algorithm 2, the performance of the differential privacy mechanism can be considered as meeting the requirements, and there is no need to reset the true output probability  $P1$  with the noise addition probability  $P2$  to end the algorithm.

In the work done in this section, we applied the method proposed in this article to the example by setting the privacy budget  $\epsilon = 1.05$  and obtained the differential Petri net model, as well as the event logs generated by this model, as shown in Table 2. Compared with the original event log, the generated event log contains all the variants of the original traces and does not generate traces that do not exist originally.

To verify the performance of the methods in this article to retain process information in the original log, we use k-anonymity as a measure of the degree of privacy of the event log and verify it. To guarantee the k-anonymity of the event log, a simple approach is to remove all traces that violate the privacy requirement according to the k-anonymity requirement, producing an event log with a large loss of process information. We set the value of k to 15, and Table 3 shows the original event log after processing and Table 4 shows the synthetic event log after processing. The event log generated by the approach in this article contains more active trace variants when the k-anonymity requirements are the same.

### **Restore Frequency Information**

The approach used in this article incorporates a randomized response approach, which protects log privacy and reduces the loss of process information, while by reversing the randomized response approach, we can restore the trace frequency information in the original logs through the processed event logs, yielding frequency results that approximate the original logs.

**Table 3.** Original event log after processing.

$s_1$	Outpatient Guidance,Registration,Ask doctor,SimpleCheck, fever detection	40
$s_3$	Outpatient Guidance,Ask doctor, Registration,SimpleCheck, fever detection	25

**Table 4.** Synthetic event log after processing.

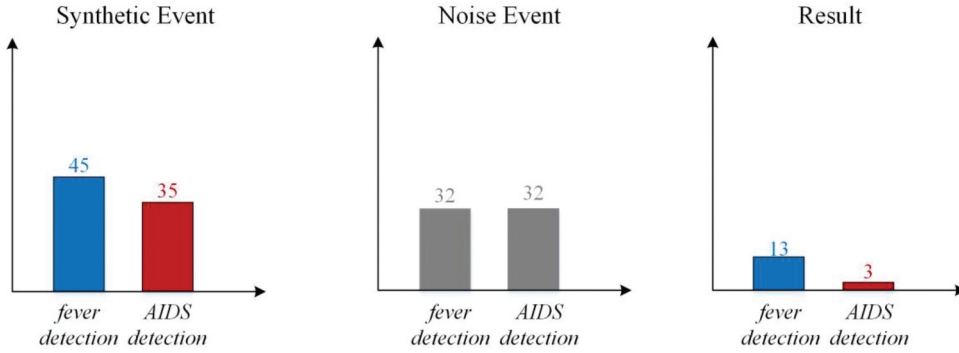
$s_1$	Outpatient Guidance,Registration,Ask doctor,SimpleCheck, fever detection	26
$s_2$	Outpatient Guidance,Registration,Ask doctor,SimpleCheck, AIDS detection	19
$s_3$	Outpatient Guidance,Ask doctor, Registration,SimpleCheck, fever detection	19
$s_4$	Outpatient Guidance,Ask doctor, Registration,SimpleCheck, AIDS detection	16

The differential privacy mechanism proposed in this article can ensure that the protected event logs still have data availability. In the randomized anonymization structure proposed in [Section 3.3](#), we set the initial probability to  $P1 = 0.2$  and  $P2 = 0.8$ . i.e., each private label AIDS detection or sensitive label fever detection has 80% probability of being injected with noise, corresponding to [Figure 8](#) The part of the red arc in the differential Petri net, and 20% probability of outputting the true label directly, corresponding to the part of the blue arc in [Figure 8](#).

Therefore, in the event log after processing, there are about 64 important labels in the related cases 80 cases are activity labels outputted after noise injection, and since the label fever detection or label AIDS detection is outputted with the same probability when noise is injected, there are about 32 label fever detection with noise injection and 32 label AIDS detection with noise injection. A total of 45 activity fever detections and 35 activity AIDS detections were recorded in [Table 2](#), and it is presumed that the number of activity fever detections output with true labels was 13 and the number of activity AIDS detections was 3. In the processed logs, it can be presumed that about 19% of the patients performed AIDS detection. According to [Table 1](#), a total of 65 event fever detections and 15 event AIDS detections were recorded in the original event log, and 23% of the patients performed AIDS detection. It can be concluded that the virtual event log generated by the differential Petri net in this article can approximate the frequency information of the activity traces in the original log, and can provide approximate results with the original log. The calculation process is shown in [Figure 10](#).

## Experimental Evaluation

In the previous section, we combined the behavioral profile with the randomized response to ensure that the noise injected into the original logs is the traces that were already present in the logs by selecting the important labels. Thus solving RQ1, which we proposed in [Section 1](#), and in RQ2, we propose to reduce the loss of process information during log processing. To evaluate the performance of the approach in this article in terms of process information reduction, in this section we performed experiments using a public dataset. In the following we refer to the proposed approach as DP-PetriNet. The details of this part are as follows: [Section 4.1](#) presents information on the dataset used



**Figure 10.** Data availability verification.

for the experiments, [section 4.2](#) describes the experimental settings in detail, [section 4.3](#) discusses the results of the experiments.

### Dataset

We verified the algorithm of this article on the open source tool PM4PY<sup>1</sup>. We conducted evaluation experiments using a public dataset BPI Challenge 2017 - Offer log<sup>2</sup>. According to Algorithm 2, DP-PetriNet processes the eligible activity traces in this event log, while the unqualified ones are left unprocessed. The information of this dataset is shown in [Table 5](#). In this experiment, we set  $P1 = 20\%$ ,  $P2 = 80\%$ , and  $\epsilon = 1.05$ . The information of the event log after processing by the approach in this article is shown in [Table 6](#).

### Experimental Settings

We compare the experimental results of DP-PetriNet with a Baseline method. This Baseline method provides privacy guarantees in the following way.

**Baseline:** Based on the definition of  $k$ -anonymity, the traces that appear less than  $k$  times in the original log  $L$  are removed to ensure the  $k$ -anonymity of Baseline log  $L_B$ .

The degree of privacy of the event log can be measured using  $k$ -anonymity as a metric, but guaranteed  $k$ -anonymity results in the loss of process information in the original log. To evaluate this, we use the number of variants in the original log retained by the approach as a metric to verify the performance of DP-PetriNet and Baseline in retaining process information in the original log when  $k$  takes the same value, respectively. In addition, while reducing the loss of process information, we also hope that the quality of the process models mined from the processed event logs will not be significantly degraded. To verify the quality of the process models, for the event logs processed by DP-PetriNet or Baseline, we use the Inductive approach to discover process

**Table 5.** Event log information.

Variant	#
<i>O_Create Offer, O_Created, O_Sent (mail and online), O_Returned, O_Accepted</i>	16299
<i>O_Create Offer, O_Created, O_Sent (mail and online), O_Returned, O_Refused</i>	3532
<i>O_Create Offer, O_Created, O_Sent (mail and online), O_Returned, O_Cancelled</i>	2393
<i>O_Create Offer, O_Created, O_Sent (online only), O_Returned, O_Accepted</i>	929
<i>O_Create Offer, O_Created, O_Sent (online only), O_Returned, O_Cancelled</i>	62
<i>O_Create Offer, O_Created, O_Sent (online only), O_Returned, O_Refused</i>	41
<i>O_Create Offer, O_Created, O_Sent (mail and online), O_Cancelled</i>	16365
<i>O_Create Offer, O_Created, O_Cancelled</i>	1203
<i>O_Create Offer, O_Created, O_Sent (mail and online), O_Refused</i>	962
<i>O_Create Offer, O_Created, O_Sent (online only), O_Cancelled</i>	875
<i>O_Create Offer, O_Created, O_Sent (mail and online),</i>	108
<i>O_Create Offer, O_Created, O_Sent (online only), O_Refused</i>	101
<i>O_Create Offer, O_Created, O_Refused</i>	59
<i>O_Create Offer, O_Created, O_Sent (mail and online), O_Returned</i>	48
<i>O_Create Offer, O_Created, O_Sent (online only),</i>	17
<i>O_Create Offer, O_Created, O_Sent (online only), O_Returned</i>	1

**Table 6.** Event log information after processing.

Variant	#
<i>O_Create Offer, O_Created, O_Sent (mail and online), O_Returned, O_Accepted</i>	10974
<i>O_Create Offer, O_Created, O_Sent (mail and online), O_Returned, O_Cancelled</i>	5448
<i>O_Create Offer, O_Created, O_Sent (mail and online), O_Returned, O_Refused</i>	5802
<i>O_Create Offer, O_Created, O_Sent (online only), O_Returned, O_Accepted</i>	582
<i>O_Create Offer, O_Created, O_Sent (online only), O_Returned, O_Cancelled</i>	248
<i>O_Create Offer, O_Created, O_Sent (online only), O_Returned, O_Refused</i>	202
<i>O_Create Offer, O_Created, O_Sent (mail and online), O_Cancelled</i>	16365
<i>O_Create Offer, O_Created, O_Cancelled</i>	1203
<i>O_Create Offer, O_Created, O_Sent (mail and online), O_Refused</i>	962
<i>O_Create Offer, O_Created, O_Sent (online only), O_Cancelled</i>	875
<i>O_Create Offer, O_Created, O_Sent (mail and online),</i>	108
<i>O_Create Offer, O_Created, O_Sent (online only), O_Refused</i>	101
<i>O_Create Offer, O_Created, O_Refused</i>	59
<i>O_Create Offer, O_Created, O_Sent (mail and online), O_Returned</i>	48
<i>O_Create Offer, O_Created, O_Sent (online only),</i>	17
<i>O_Create Offer, O_Created, O_Sent (online only), O_Returned</i>	1

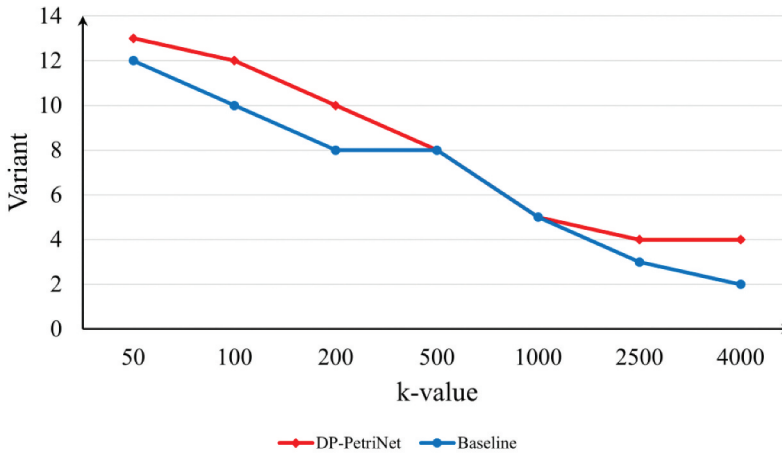
models from these logs. Afterward, we measured the quality of the models by the Fitness (Adriansyah, van Dongen, and van der Aalst 2011) and Precision (Adriansyah et al. 2015) of the discovered models relative to the original event logs L. In addition, we evaluate the effect of different k values on model quality in our experiments.

## Results

The experimental results are shown in Table 7. Figure 11 shows the retention of process trace variants in the original event log at different k values. It can be found that the number of variants contained in the processed event logs of both DP-PetriNet and Baseline approaches decreases as the value of k increases. However, in most cases, the DP-PetriNet approach retains more variants than the Baseline approach, and only in the case of some k values, for example, when the value of k is taken in the range (500,1000), the number of

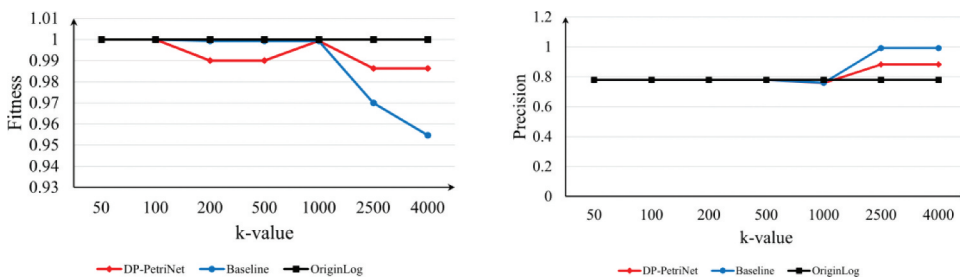
**Table 7.** Experimental data.

	Variant		Fitness		Precision	
	DP-PetriNet	Baseline	DP-PetriNe	Baseline	DP-PetriNe	Baseline
k=50	<b>13</b>	12	1	1	0.78	0.78
k=100	<b>12</b>	10	1	1	0.78	0.78
k=200	<b>10</b>	8	0.99	<b>0.9993</b>	0.78	0.78
k=500	8	8	0.99	<b>0.9993</b>	0.78	0.78
k=1000	5	5	0.9993	0.9993	0.7576	<b>0.7596</b>
k=2500	<b>4</b>	3	<b>0.9863</b>	0.97	0.8831	<b>0.9921</b>
k=4000	<b>4</b>	2	<b>0.9863</b>	0.9547	0.8831	<b>0.9921</b>

**Figure 11.** Number of retained trace variants.

variants retained by the DP-PetriNet approach is comparable to that of the Baseline approach. This result indicates that the DP-PetriNet approach always retains more or a considerable number of variants compared to the Baseline approach for the same k value, in other words, the DP-PetriNet approach always retains more process information in the original log for the same k value, reducing the loss of process information in the original log.

Figure 12 shows the Fitness and Precision between the process model mined from the processed event logs using Inductive approach and the original event logs. Also, we include the results of the model mined from the unprocessed event logs as a comparison. It can be found that, compared with the Baseline approach, the Precision of the DP-PetriNet approach is comparable to that of the Baseline approach when the value of k is taken small, such as k = 50 or k = 100; and when the value of k is taken large, such as k = 2500, the Precision of the DP-PetriNet is 0.8831 at this time. The Precision of Baseline method is 0.9921, and the performance of DP-PetriNet will be slightly lower than that of Baseline approach at this time, which is due to the fact that DP-PetriNet approach retains more kinds of activity traces compared with Baseline approach at k = 2500.



**Figure 12.** Fitness and precision.

In terms of Fitness measure, compared with Baseline approach, when the value of  $k$  is small, the Fitness of DP-PetriNet approach is basically comparable to Baseline approach, for example, when  $k$  is 200, the Fitness of DP-PetriNet is 0.99, and the Fitness of Baseline approach is 0.9993; and when  $k$  takes a larger value, such as  $k = 2500$ , the Precision of DP-PetriNet is 0.9863 and the Precision of Baseline approach is 0.97, and the performance of DP-PetriNet is slightly better than that of Baseline approach.

The experimental results demonstrate that the performance of the DP-PetriNet approach proposed in this article is slightly lower in Precision and higher in Fitness compared to the Baseline approach in terms of Precision and Fitness metrics. This result indicates that for the same  $k$  value, the event logs processed by the Baseline approach and the DP-PetriNet approach have their own advantages in terms of quality metrics for the models mined using process mining techniques, such as higher Precision for the Baseline approach and higher Fitness for the DP-PetriNet approach. Overall, the quality of the models mined by both approaches is comparable. Combining the number of variants in the original logs retained by the two approaches, we can conclude that the DP-PetriNet approach can retain more process variants in the original logs compared with the Baseline approach with comparable quality of the mined models, which means that the loss of process information in the original logs is reduced.

## Conclusion

In this article, we propose a differential algorithm based on behavioral profile and randomized response to build differential Petri nets that solve the two research questions presented in [Section 1](#). The algorithm is secure provable, performance usable and decision interpretable. It selects the important labels in the event logs by the behavioral profile between activities, injects the important labels as noise into the Petri net model mined from the original event logs through a randomized response approach, and

builds a differential Petri net model, which is used to synthesize the processed event logs.

Compared with the existing literature, this article has the following potential innovations: (1) In terms of content, different from the existing works based on the privacy protection perspective alone, this article integrates the knowledge of privacy protection and Petri net domain, combines the behavioral profile theory with the differential privacy approach, and the research perspective is more comprehensive and deeply. (2) In terms of approach, existing research focuses on processing the original logs and obtaining differential privacy event logs by injecting noise or merging similar information. In this paper, we consider the loss of process information in the processed event logs, and propose a approach that balances privacy and process information. In this paper, we consider the loss of process information in the processed event logs, and propose a approach that balances privacy and process information. It provides a supplement to the research in the area of differential privacy protection of event logs.

The differential Petri nets established by this approach are differential privacy compliant. According to the post-processing definition of differential privacy, the synthesized event logs are also differential privacy compliant, and since the important labels as noise injection exist in the original event logs, RQ1 is solved and the traces contained in the processed event logs are all present in the original logs. At the same time, we solve RQ2 by introducing a random response mechanism that retains more process variants of the original log under the same k-anonymity requirement, thus reducing the loss of process information from the original log. In addition, the approach in this article also effectively restores the frequency information of process variants in the original logs. The performance of the approach is experimentally demonstrated on a public dataset.

## Notes

1. <https://pm4py.fit.fraunhofer.de/documentation>.
2. [https://data.4tu.nl/articles/dataset/BPI\\_Challenge\\_2017\\_-\\_Offer\\_log/12705737](https://data.4tu.nl/articles/dataset/BPI_Challenge_2017_-_Offer_log/12705737).

## Acknowledgements

We also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

## Disclosure Statement

No potential conflict of interest was reported by the authors.



## Funding

Supported by the National Natural Science Foundation, China [No. 61572035, 61402011], Key Research and Development Program of Anhui Province [2022a05020005], the Leading Backbone Talent Project in Anhui Province, China [2020-1-12], and the Open Project Program of the Key Laboratory of Embedded System and Service Computing of Ministry of Education [No.ESSCKF2021-05].

## ORCID

Daoyu Kan  <http://orcid.org/0000-0003-0315-9503>  
 Xianwen Fang  <http://orcid.org/0000-0001-8531-7215>  
 Ziyong Gong  <http://orcid.org/0000-0003-1724-1364>

## References

- Adriansyah, A., J. Munoz-Gama, J. Carmona, B. F. van Dongen, and W. M. P. van der Aalst. 2015. Measuring precision of modeled behavior. *Information Systems and E-Business Management* 13 (1):37–67. doi:10.1007/s10257-014-0234-7.
- Adriansyah, A., B. F. van Dongen, and W. M. P. van der Aalst. 2011. Conformance checking using cost-based fitness analysis. In *2011 IEEE 15th International Enterprise Distributed Object Computing Conference*, 55–64. doi:10.1109/EDOC.2011.12
- Augusto, A., R. Conforti, M. Dumas, M. L. Rosa, F. M. Maggi, A. Marrella, M. Mecella, and A. Soo. 2019. Automated discovery of process models from event logs: Review and benchmark. *IEEE Transactions on Knowledge and Data Engineering* 31 (4):686–705. doi:10.1109/TKDE.2018.2841877.
- Batista, E., and A. Solanas. 2021. A uniformization-based approach to preserve individuals' privacy during process mining analyses. *Peer-To-Peer Networking and Applications* 14 (3):14. doi:10.1007/s12083-020-01059-1.
- Dwork, C. 2008. Differential privacy: A survey of results. In *Theory and applications of models of computation*, ed. M. Agrawal, D. Du, Z. Duan, and A. Li, 1–19. Berlin Heidelberg: Springer.
- Elkoumy, G., S. Fahrenkrog-Petersen, M. Dumas, P. Laud, A. Pankova, and M. Weidlich. 2022. Shareprom: A tool for privacy-preserving inter-organizational process mining, August 5.
- Elkoumy, G., S. A. Fahrenkrog-Petersen, M. F. Sani, A. Koschmider, F. Mannhardt, S. N. Von Voigt, M. Rafiei, and L. V. Waldthausen. 2021. Privacy and confidentiality in process mining: Threats and research challenges. *ACM Transactions on Management Information Systems* 13 (1):1–11. doi:10.1145/3468877.
- Elkoumy, G., A. Pankova, and M. Dumas. 2021. Mine me but don't single me out: Differentially private event logs for process mining. In *2021 3rd International Conference on Process Mining (ICPM)*, 80–87. doi:10.1109/ICPM53251.2021.9576852
- Fahrenkrog-Petersen, S. 2019. Providing privacy guarantees in process mining, 23–30.
- Fahrenkrog-Petersen, S. A., H. van der Aa, and M. Weidlich. 2019. PRETSA: Event log sanitization for privacy-aware process discovery. In *2019 International Conference on Process Mining (ICPM)*, 1–8. doi:10.1109/ICPM.2019.00012
- Fahrenkrog-Petersen, S. A., H. van der Aa, and M. Weidlich. 2020. PRIPEL: Privacy-preserving event log publishing including contextual information. In *Business Process Management*, ed. D. Fahland, C. Ghidini, J. Becker, and M. Dumas, 111–28. Cham, Switzerland: Springer International Publishing.

- Feng, P., H. Zhu, Y. Liu, Y. Chen, and Q. Zheng. 2018. Differential privacy protection recommendation algorithm based on student learning behavior. In *2018 IEEE 15th International Conference on E-Business Engineering (ICEBE)*, 285–88. doi:[10.1109/ICEBE.2018.00054](https://doi.org/10.1109/ICEBE.2018.00054)
- Hou, Y., X. Xia, H. Li, J. Cui, and A. Mardani. 2022. Fuzzy differential privacy theory and its applications in subgraph counting. In *IEEE Transactions on Fuzzy Systems*, 1–1. doi:[10.1109/TFUZZ.2022.3157385](https://doi.org/10.1109/TFUZZ.2022.3157385)
- Kabierski, M., S. A. Fahrenkrog-Petersen, and M. Weidlich. 2021. Privacy-aware process performance indicators: Framework and release mechanisms. In *Advanced information systems engineering*, ed. M. L. Rosa, S. Sadiq, and E. Teniente, 19–36. Cham, Switzerland: Springer International Publishing.
- Leemans, S. J. J., D. Fahland, and W. M. P. van der Aalst. 2013. Discovering block-structured process models from event logs—A constructive approach. In *Application and theory of petri nets and concurrency*, ed. J.-M. Colom and J. Desel, 311–29. Berlin Heidelberg: Springer.
- Liu, C. 2022. Formal modeling and discovery of multi-instance business processes: A cloud resource management case study. *IEEE/CAA Journal of Automatica Sinica* 9 (12):2151–60. doi:[10.1109/JAS.2022.106109](https://doi.org/10.1109/JAS.2022.106109).
- Liu, C., L. Cheng, Q. Zeng, and L. Wen. 2022. Formal modeling and discovery of hierarchical business processes: A petri net-based approach. In *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 1–12. doi:[10.1109/TSMC.2022.3195869](https://doi.org/10.1109/TSMC.2022.3195869)
- Liu, C., H. Li, S. Zhang, L. Cheng, and Q. Zeng. 2022. Cross-department collaborative healthcare process model discovery from event logs. In *IEEE Transactions on Automation Science and Engineering*, 1–11. doi:[10.1109/TASE.2022.3194312](https://doi.org/10.1109/TASE.2022.3194312)
- Mannhardt, F., A. Koschmider, N. Baracaldo, M. Weidlich, and J. Michael. 2019. Privacy-preserving process mining: Differential privacy for event logs. *Business & Information Systems Engineering* 61 (5):595–614. doi:[10.1007/s12599-019-00613-3](https://doi.org/10.1007/s12599-019-00613-3).
- Mannhardt, F., S. A. Petersen, and M. F. Oliveira. 2018. Privacy challenges for process mining in human-centered industrial environments. In *2018 14th International Conference on Intelligent Environments (IE)*, 64–71. doi:[10.1109/IE.2018.00017](https://doi.org/10.1109/IE.2018.00017)
- Núñez von Voigt, S., S. Fahrenkrog-Petersen, D. Janssen, A. Koschmider, F. Tschorsch, F. Mannhardt, O. Landsiedel, and M. Weidlich. 2020. Quantifying the re-identification risk of event logs for process mining: empirical evaluation paper, 252–67). doi:[10.1007/978-3-030-49435-3\\_16](https://doi.org/10.1007/978-3-030-49435-3_16)
- Pika, A., M. T. Wynn, S. Budiono, A. H. M. Ter Hofstede, W. M. P. van der Aalst, and H. A. Reijers. 2020. Privacy-preserving process mining in healthcare. *International Journal of Environmental Research and Public Health* 17 (5):1612, Article 5. doi:[10.3390/ijerph17051612](https://doi.org/10.3390/ijerph17051612).
- Rafiei, M., L. von Waldthausen, and W. M. P. van der Aalst. 2020. Supporting confidentiality in process mining using abstraction and encryption. In *Data-driven process discovery and analysis*, ed. P. Ceravolo, M. van Keulen, and M. T. Gómez-López, 101–23. Springer International Publishing. doi:[10.1007/978-3-030-46633-6\\_6](https://doi.org/10.1007/978-3-030-46633-6_6).
- Rafiei, M., M. Wagner, and W. M. P. van der Aalst. 2020. TLKC-privacy model for process mining. In *Research challenges in information science*, ed. F. Dalpiaz, J. Zdravkovic, and P. Loucopoulos, 398–416. Cham, Switzerland: Springer International Publishing.
- Rösel, F., S. Fahrenkrog-Petersen, H. van der Aa, and M. Weidlich. 2022. A distance measure for privacy-preserving process mining based on feature learning 73–85. doi:[10.1007/978-3-030-94343-1\\_6](https://doi.org/10.1007/978-3-030-94343-1_6).
- Stefanini, A., D. Aloini, E. Benevento, R. Dulmin, and V. Mininno. 2018. Performance analysis in emergency departments: A data-driven approach. *Measuring Business Excellence* 22 (2):130–45. doi:[10.1108/MBE-07-2017-0040](https://doi.org/10.1108/MBE-07-2017-0040).

- Sweeney, L. 2002. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10 (5):557–70. doi:[10.1142/S0218488502001648](https://doi.org/10.1142/S0218488502001648).
- van der Aalst, W. 2012. Process mining: Overview and opportunities. *ACM Transactions on Management Information Systems* 3 (2):1–17. doi:[10.1145/2229156.2229157](https://doi.org/10.1145/2229156.2229157).
- van der Aalst, W. 2016. Data science in action. In *Process mining: Data science in action*, 3–23. Berlin Heidelberg: Springer. doi:[10.1007/978-3-662-49851-4\\_1](https://doi.org/10.1007/978-3-662-49851-4_1).
- van der Aalst, W., T. Weijters, and L. Maruster. 2004. Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering* 16 (9):1128–42. doi:[10.1109/TKDE.2004.47](https://doi.org/10.1109/TKDE.2004.47).
- Voss, W. G. 2016. European union data privacy law reform: General data protection regulation, privacy shield, and the right to delisting. *The Business Lawyer* 72 (1):221–34.
- Weidlich, M., A. Polyvyanyy, N. Desai, and J. Mendling. 2010. Process compliance measurement based on behavioural profiles. In *Advanced information systems engineering*, ed. B. Pernici, 499–514. Springer. doi:[10.1007/978-3-642-13094-6\\_38](https://doi.org/10.1007/978-3-642-13094-6_38).