Journal of Big Data

RESEARCH Open Access

Privacy preserved incremental record linkage

Shahidul Islam Khan[1,2*], Abir Bin Ayub Khan[1] and Abu Sayed Md Latiful Hoque[2]

*Correspondence:
nayeemkh@gmail.com

[1] Department of CSE, International Islamic University Chittagong, Chittagong, Bangladesh
[2] Department of CSE, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh

Abstract

Using an incremental approach to solve the record linkage problem is a relatively new research area. In incremental record linkage, every inserted record is compared with some existing clusters of records based on its blocking key value. Then, considering similarity, either the record will be put into an existing cluster, or a new cluster will be created for it. Although few papers have presented their solutions for incremental record linkage targeting the linkage quality or efficiency, privacy issue regarding the approach has not yet been discussed. Privacy is a major concern when record linkage is performed for sensitive data, e.g., health records, financial records, etc. In this regard, we have come up with a novel concept privacy-preserving incremental record linkage (PPiRL) which encapsulates privacy-preserving techniques with an incremental record linkage approach. In this chapter, we have proposed an end-to-end framework as our solution for PPiRL. For preserving privacy, we have used two types of privacy techniques namely phonetic encoding and generalization. We have used a recently developed phonetic algorithm "nameGist" to handle text-based features. For generalization, we have used the K-anonymization algorithm for numeric and categorical features. For handling incremental updates and internal linkage, we have used the Naive incremental clustering approach using Hierarchical Agglomerative clustering as the base clustering algorithm. We have performed various experiments to test the privacy and linkage quality of PPiRL. We have compared our work with the existing incremental record linkage framework and also with existing privacy-preserved record linkage techniques. It is apparent from our results that other than a small trade-off in linkage quality, our framework works better as a combined package of privacy and linkage solutions that any existing frameworks do not yet provide.

Keywords: Record linkage, Data matching, Privacy, Incremental, Big data

Introduction

Nowadays, fast-growing datasets that contain hundreds of millions of records are being collected, stored, processed, analyzed, and mined. To enable an in-depth analysis of such large datasets, information from multiple data sources is often required to be integrated. For getting maximum insight from integrated data (e.g., correlations among diseases in the case of a medical dataset), record linkage is essential. Record linkage is the process of identifying record pairs from different information systems that belong to the same real-world entity, i.e., a customer, or a patient. Given two repositories of records, the

record linkage process consists of determining all pairs that are similar to each other. Record linkage faces two challenges on the edge of big data. First, the high velocity of data updates swiftly makes previous linkage results extinct. Second, a massive volume of data requires a long time for applying record linkage in the traditional (batch linkage) approach. These two challenges require an incremental solution so that when data updates appear, we can swiftly update linkage results [18].

Usually, distinct identifiers, e.g., primary keys, are not always present in the databases that need to be linked. This makes record linkage a problematic task. So, to perform linkage, the common attributes of datasets are used in many cases. These include the name, birth date, address, and other personal details of an entity. Currently, maintaining privacy and confidentiality are significant challenges for record linkage. During the linking of databases across organizations using personal information, careful protection of the privacy of this information is a must. The process of discovering records of similar individuals from separate databases without disclosing identifying attributes of these individuals is known as 'privacy-preserved linkage of records,' 'linkage of blind data,' or the 'private linkage of records' problem [41].

Although few papers such as [8, 18, 29, 32, 38, 44, 45] have presented their solutions for incremental record linkage targeting the linkage quality or efficiency, the privacy issue of this approach is yet to discuss. In this regard, we have come up with a new idea called "privacy-preserving incremental record linkage" or in short "PPiRL," which encapsulates privacy-preserving techniques with an incremental approach to record linkage problems. We have the following key contributions to this research. We propose the concept of Privacy Preserved Incremental Record Linkage (PPiRL) for the first time and derived the required mathematical model for it. We also develop a PPiRL framework and finally implemented it. We test the performance of the PPiRL framework and compare it with the existing state-of-the-art techniques.

Importance of privacy preservation

Information privacy of an individual or organization deals with the ability to determine what data in a computer system to be shared with others. It is considered an important aspect of information sharing wherever personally identifiable information is accumulated in any form. Depending on the category of information, e.g., health, finance, etc. sometimes it is more important to maintain privacy than others.

Medical databases, in particular, involve highly private information. Detailed information about patients might become obtainable when databases are linked, such as some people having certain chronic diseases and also having financial problems. Security weaknesses of a healthcare information system can result in privacy losses as Protected Health Information (PHI) of patients has high sell values in the underground markets. It is an increasing target for hackers to break down the security of health systems and expose the private data of patients for money.

Nobody likes his medical records to be revealed to unauthorized persons. Hackers or eavesdroppers generally aim to exploit personally identifiable information to do business with the extracted information or extort a famous person or celebrity. A social security number (SSN) is sold for twenty-five cents, and a credit card number can be priced at one dollar in the United States underground market. Surprisingly, a medical record's

price in the same market ranges from ten dollars up to one thousand dollars. These sold records normally have a date of birth, the name of the patient, numbers of health policy, codes of diagnosis, and other significant ID numbers. Eavesdropper uses this data for creating duplicated IDs to buy health-related products, insurance, drugs, etc. From the year of 2014, the extent of hacking over healthcare servers has increased by a significant margin. The attackers' motivation is to get huge PHI in a single successful hack. Analyzing the data provided by the U.S.Department of Health and Human Services, it is found that hackers are increasingly targeting healthcare servers which is very alarming to the health information systems using record linkage [23, 25, 27, 36].

On the other hand, the banking industry always looks for new methods which help to create more precise customer profiles. For example, exploitation of internal and external data, even aggregate information about customers' purchasing habits to other organizations using record linkage. These methods help enhance customer credibility. However, there are potential risks to violate financial ethics which can occur as data abuse. So, in the case of financial records, privacy preservation is vital [17, 30].

Application areas of PPiRL

Privacy needs careful consideration when data from several organizations are linked. Many fields like public health research, health surveillance, census, and centralized data warehouses are in constant need of privacy-preserved linkage as there are many parties involved in the linkage process.

In public health research, researchers often set on to investigate the types of injuries caused by car accidents, intending to uncover the correlation between types of accidents and the resulting injuries [41]. This kind of research can have a significant influence on potentially lifesaving changes in policymaking. In this scenario, several parties such as hospitals, the police, as well as public and private health insurers are involved.

Financial organizations such as online marketplaces, e-commerce sites, and banks require to develop a complete and up-to-date profile of their customers by linking data from different organizations. Here also several financial institutions such as banks and e-commerce sites are involved [17].

In health surveillance, early outbreak detection systems to prevent infectious diseases require data from various sources to be gathered and linked, such as human health data, animal health data, and consumed drugs data. Privacy is a prime concern when such data are linked and stored at a central location [27, 40].

Our contributions

We have three major contributions which are apparent throughout the paper.

1. To the best of our knowledge, We are the first to recognize privacy-preserving incremental record linkage as a new field of research. Recognition of this field paves the way for solving the problems of record linkage, integration, and data mining relating to the volume and velocity of data along with privacy issues.
2. We have proposed a novel end-to-end framework that encompasses both privacy and linkage of data in an incremental approach. We have named it privacy-preserv-

ing incremental record linkage (PPiRL). We have also provided the necessary definition and mathematical model for PPiRL.

3. We have implemented our PPiRL framework and provide various comparisons of our framework with traditional privacy-preserving record linkage (PPRL) techniques and incremental record linkage (IRL) techniques.

Organization of the paper

We have organized the paper as follows. In the "Literature review" section, we provided a review of the important literature related to record linkage, privacy-preserving record linkage, and incremental record linkage. We have provided some background and formulated the problem of privacy-preserving incremental record linkage in the "Background knowledge and problem formulation" section. Our main contribution, the PPiRL framework, is explained next. Experimental results, privacy evaluation, and comparisons are presented in "Experimental Results" section. We have presented a short discussion on the result in the next section. Finally, the "Summary" section concludes this paper.

Literature review

Record linkage

Record linkage or entity resolution refers to the process of identifying and aggregating records from one or more datasets, which represent the same real-world entities. Recently the world has encountered an eruption in the volume and velocity of data that is being accumulated by individuals as well as organizations. All these data are either generated by the people or about the people. To achieve good results in data mining, the quality of the data is essential. A serious obstacle to proper data analysis is the noise in the collected data [3, 28, 35]. Low-quality data which contain erroneous, missing data, or out-of-date values generate low accuracy outcomes after data analysis [9]. In order to improve the quality of data and do complex data analysis and mining, a solution is to integrate data from different sources. This integration of data paves the way to identify conflicting data values, enrich data, or impute missing values [22]. A traditional approach to getting the linkage of records is to find the similarity among record pairs. After similarity calculation, supervised or unsupervised algorithms can be applied to extract the linkage result.

Record linkage [16, 22], schema matching [6], and data fusion [6, 7] are the three main tasks in data integration. Among them, record linkage is aimed at identifying all records that refer to similar real-world entities in several databases. It can also be applied to detect identical records in a single database [15, 33]. For record linkage, three significant complications can be recognized. Firstly, linkage quality plays a crucial role in the linkage of records. As the fact that real-world data are 'dirty' is responsible for the loss of quality of linkage [21]. Only the exact matching of personal identifying features is not able to give us the desired output. In this case, we need approximate matching besides accurate matching to achieve good accuracy in linkage quality [9, 14]. To decrease the number of potential comparisons required between scalability is essential. The usage of expensive similarity comparison methods creates a performance bottleneck [4, 12]. This challenge can be overcome by using proper indexing techniques [11].

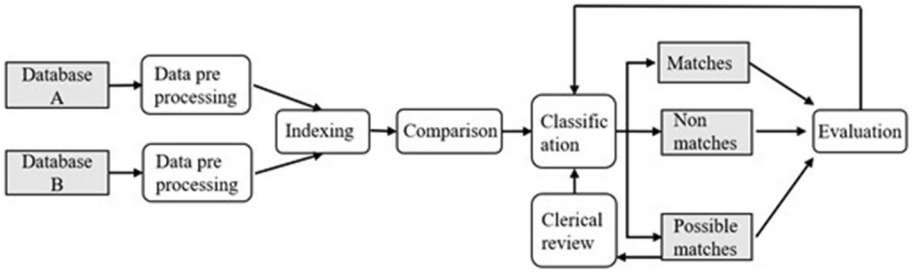

Fig. 1 Outline of the general record linkage process

Figure 1 illustrates the outline of the general record linkage process comprising several steps. Data preprocessing which includes data cleaning and standardization is the first step in this process. It is a crucial step because real-world data contain inconsistent, noisy data [3, 35]. Indexing [11] is the second step in the process. In the comparison step, record pairs are compared elaborately with the help of similarity functions [10]. The classification step classifies the record pairs using a decision model thus generating matches, non-matches, or possible matches [19]. Finally, evaluation measures of different types are deployed to measure the complexity [11] and quality of resultant linkage [12].

Privacy-preserving record linkage (PPRL)

Definition 1 *Privacy-preserving record linkage (PPRL):* Assume $P_1, \ldots, P_m$ are the m owners of the databases $D_1, \ldots, D_m$, respectively. They wish to determine which of their records $R_1^i \in D_1$, $R_2^j \in D_2$,..., $R_m^k \in D_m$ match based on their demographic data according to a decision model $C(R_1^i, R_2^j, \ldots, R_m^k)$ that classifies records of different datasets into one of the two classes: M (Match), and N(Non-match). $P_1, \ldots, P_m$ do not wish to reveal their actual records $R_1^i, R_2^j, \ldots, R_m^k$ with any other party. They however are prepared to disclose to each other, or to an external party, the actual values of some selected attributes of the records that are in class M to allow analysis [40]. A viable PPRL solution that can be used in real-world applications should have three properties: scalability, linkage quality, and privacy.

Figure 2 illustrates the outline of the privacy preserving record linkage process. Large databases across organizations needed to be linked. At the same time preserving the privacy of the records stored in these databases is also crucial. This necessity directs a new research area called privacy-preserving record linkage (PPRL) [13, 39, 42]. PPRL is alternatively called as privacy record linkage [1, 2, 24, 46] and blind-folded record linkage [13, 43]. Due to privacy concerns, commercial interests, or legal restrictions, it is often not allowed to exchange private or confidential data between organizations. When there arises a cross-organizational project, databases of different organizations need to link in such a way that no sensitive information is being exposed to any of the parties involved, and no outsider can eavesdrop on the data to learn anything. PPRL ensures that after the end of a linkage project, only a limited amount of information is disclosed to the exchanging parties. The disclosed information may contain, (i) the number of records

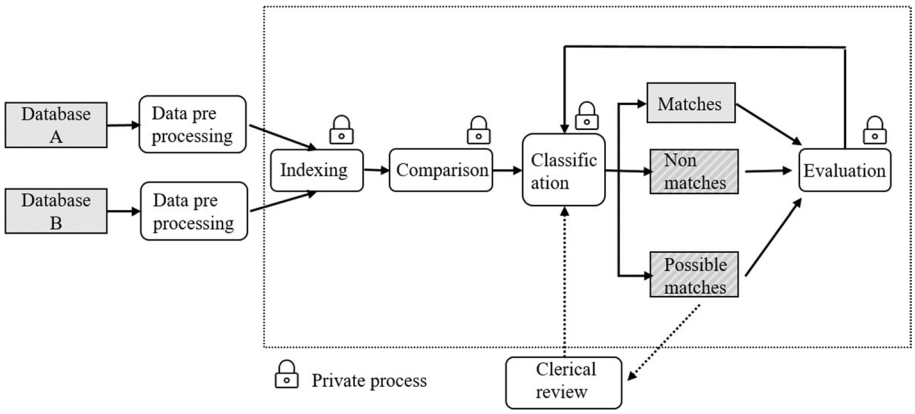

Fig. 2 Outline of the general privacy-preserving record linkage process adopted from [40]

that have been classified as matches, (ii) the attributes of these matched records, and (iii) a selected set of attributes from these matched records [40].

Incremental record linkage

Incremental record linkage (IRL) is the clustering process where only the newly arrived records will be compared with existing clusters. Then, based on similarity, either the new records will be put into some existing cluster(s), or a new cluster will be created for it if the new records are dissimilar to existing clusters according to some threshold value. Incremental record linkage has been studied in [44, 45]; however, they focused on evolving matching rules and discussed concisely only evolving data.

On the other hand, incremental graph clustering methods have been proposed by some researchers. Mathieu et al. [29] studied incremental correlation clustering for the following two cases: (1) one vertex is added each time, and (2) already identified clusters need to be preserved. Charikar et al. [8] studied incremental clustering when the number of clusters is predefined. Both papers focused on theoretical analysis rather than implementations. A novel incremental heuristic algorithm was presented in [38] for the Clique Partition Problem (CPP), a well-studied graph partitioning problem. The algorithm was much faster for the tested datasets comparing the batch linkage algorithm. Privacy issues were not considered in any of the above research.

An efficient approach for incremental record linkage has been proposed in [18] where the authors presented a framework using several algorithms and showed viable efficiency compared to the previous works. Nasciment et al. [32] proposed heuristic-based approaches to speed up the performance of the IRL algorithm. Both papers deal with linkage quality and efficiency. None of them considers the privacy issues for record linkage. To the best of our knowledge, our framework is the first to perform an incremental linkage that considers privacy issues.

Background knowledge and problem formulation

Some key terms related to incremental record linkage are discussed below.

Base dataset: A large collection of database records having both identifiable and non-identifiable attributes denoted by D here.

Increment: A dataset that contains records that need to be merged with the base dataset denoted by ΔD.

Batch record linkage: Here, for each increment dataset ΔD, the record linkage process is executed for the whole dataset D+ΔD. Let us assume a scenario in which our base dataset contains one million records whereas each increment contains one thousand records. In batch record linkage, we have one million as our starting point to apply to the cluster. When the first increment arrives, we have to perform clustering over one million and one thousand combined records. It is a time-consuming process and hence inefficient. The process is illustrated in Fig. 3.

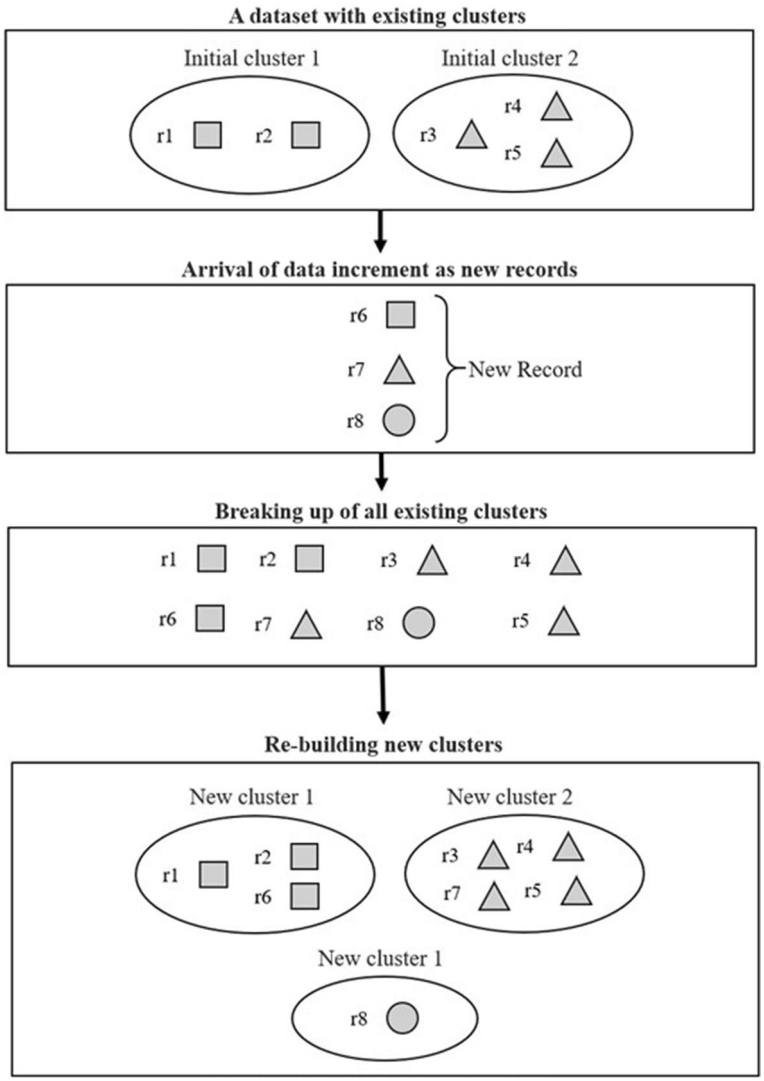

Fig. 3 Stages of batch record linkage

The figure is divided into four boxes where an arrow indicates the direction of one box to another. Each of the boxes represents a distinct stage in the batch record linkage process.

Incremental record linkage (IRL): An incremental record linkage process preserves the clusters developed from the base dataset D and merges the records from the incremental dataset ΔD using a similarity function. The IRL process creates new clusters if some of the records ofΔD are not similar to any of the existing clusters based on a similarity function. In order to get a practical understanding of how incremental record linkage works, Fig. 4 will be helpful. Here, three boxes in the figure represent three distinct stages of the IRL process.

Definition 2 *Incremental Record Linkage (IRL):* Let D be a set of records and ΔD be an increment to D. Let ρ_D be the clustering of records in D. Incremental record linkage clusters records in D + ΔD based on ρ_D. We denote the incremental record linkage method by f and denote the results by f(D, ΔD, ρ_D).

The aim of incremental record linkage is to improve performance significantly compared to its corresponding batch linkage algorithm especially if the increment is small [18]. Specifically, the computation of f(D, ΔD, ρ_D) should be faster than the computation of F(D + ΔD) if $|\Delta D| << |D|$ holds. At the same time, incremental record linkage should

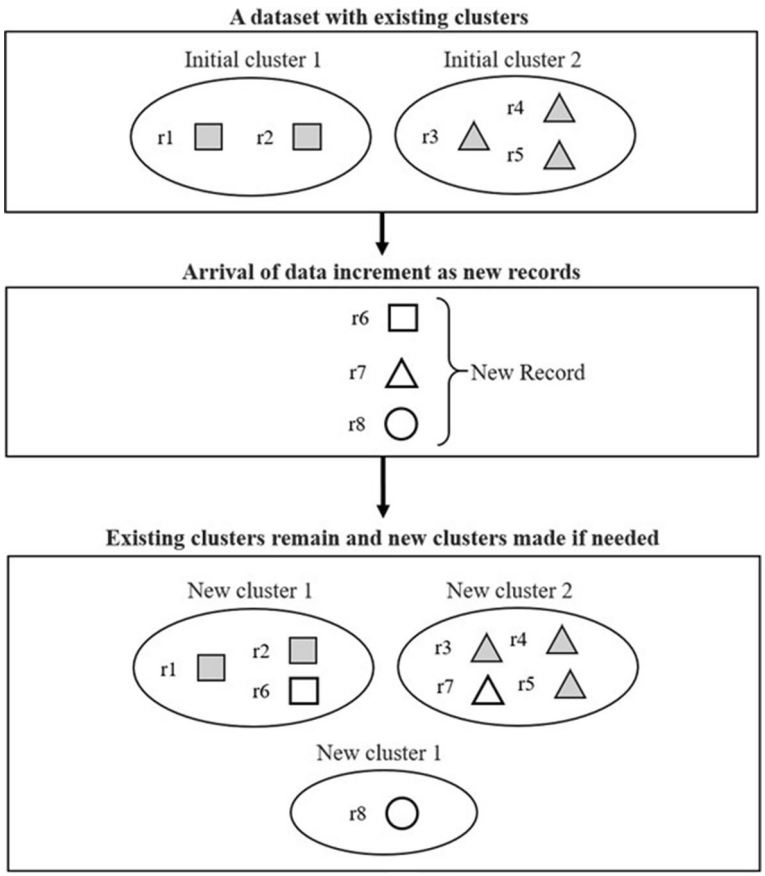

Fig. 4 Stages of incremental record linkage

achieve equivalent accuracy to its reference batch algorithm. We denote this constraint as f(D, ΔD, ρ_D) $\approx F(D + \Delta D)$.

Now we formally define the problem of privacy-preserving incremental record linkage. For a set of records, privacy-preserving incremental record linkage is essentially a combination of linkage and privacy preservation. In this problem, each cluster generally contains records where privacy is ensured with the help of several privacy-preserving techniques. The records of the cluster represent a distinct real-world entity. The linkage should have both a high recall value and a high precision value.

Definition 3 *Privacy-Preserving Incremental Record Linkage (PPiRL):* Let D be a set of records and A is the set of attributes of D. Let $\bar{A}$ is the set of sensitive attributes of D and $\bar{A} \subset A$. ΔD is an increment to D. We denote the privacy preservation method by Γ and denote the privacy preserved results $\Gamma(D)$ by $\bar{D}$, and $\Gamma(\Delta D)$ by $\bar{\Delta D}$ respectively. Let $\bar{\rho_D}$ be the clustering of records in $\bar{D}$. Privacy-preserving Incremental record linkage clusters records in $\bar{D} + \bar{\Delta D}$ based on $\bar{\rho_D}$. We denote the privacy-preserved incremental record linkage method by f', and denote the results by $f'(\bar{D}, \bar{\Delta D}, \bar{\rho_D})$.

Privacy preserving incremental record linkage (PPiRL) has three goals. PPiRL wants to ensure the privacy of sensitive records. it aims to improve performance significantly compared to the corresponding privacy-preserving batch clustering algorithm. Specifically, the computation of $f'(\bar{D}, \bar{\Delta D}, \bar{\rho_D})$ should be faster than the computation of F($\bar{D} + \bar{\Delta D}$) if $|\bar{\Delta D}| << |\bar{D}|$ holds. On the other hand, PPiRL tries to achieve equivalent accuracy to its reference batch algorithm. We denote this constraint as $f'(\bar{D}, \bar{\Delta D}, \bar{\rho_D}) \approx$ F($\bar{D} + \bar{\Delta D}$).

PPiRL, an end-to-end Framework

Our proposed solution is an end-to-end framework for record linkage which considers a significant reduction of time for performing record linkage along with privacy preservation without compromising the linkage quality. There are five basic steps in the framework with different functionality. Data pre-processing, privacy preservation, blocking, clustering, and evaluation are the five stages of our framework illustrated for the base dataset in Fig. 5 and for increments in Fig. 6.

Data pre-processing

Pre-processing of data helps improve the condition of data by handling errors and inconsistencies from data. Although data quality issues are found in a single dataset, quality issues become serious when data is integrated from multiple sources into a warehouse [35]. Some essential steps of data pre-processing are feature selection, data standardization, data cleaning, missing data imputation, normalization, etc. Details of data preprocessing are out of the scope of this paper.

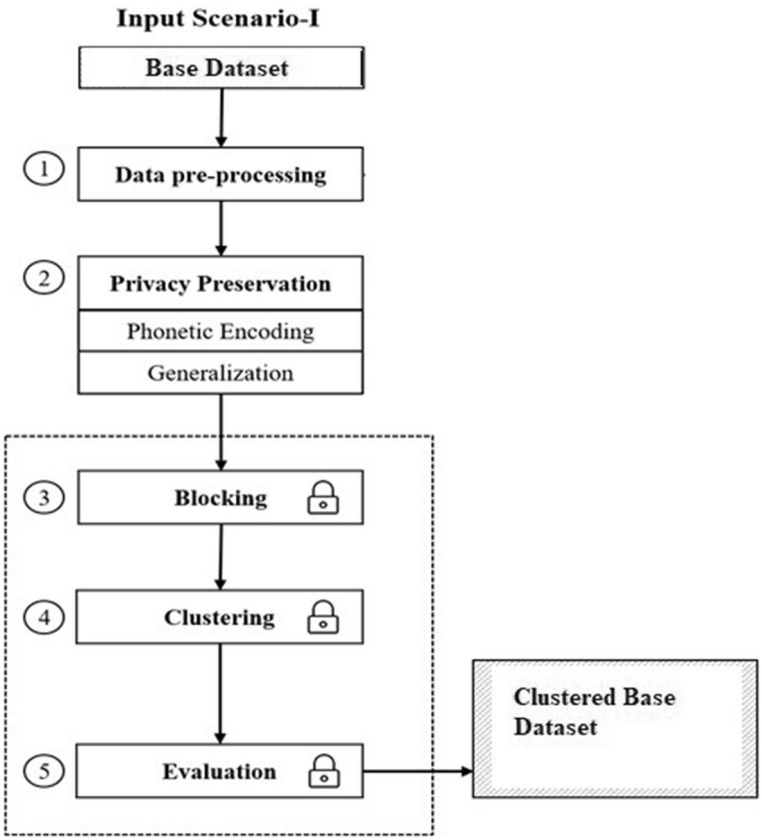

Fig. 5 PPIRL, an end-to-end framework steps for the base dataset

Privacy preservation

Depending on the attributes which are more prevalent in the healthcare datasets we have selected two types of privacy techniques to be implemented in our framework. To best suit our purpose at times we have used state-of-the-art algorithms.

Phonetic encoding

The phonetic encoding algorithm groups values that have similar pronunciations. It inherently provides privacy and increases scalability as well. The procedure is illustrated in Fig. 7. One of the most sensitive attributes in healthcare datasets is the name of the patients. A leak of a patient's name could jeopardize the patient's privacy in a bad way. Using phonetic encoding we got the following advantages:

1. Names will be encoded. So they can not be easily identified.
2. Names will be generelized. That is similar pronouncing names with different spelling will produce same code.
3. Because of generalization of names, the output code is robust against noises and spelling errors, which are common in healthcare centers specially in the developing countries [26].

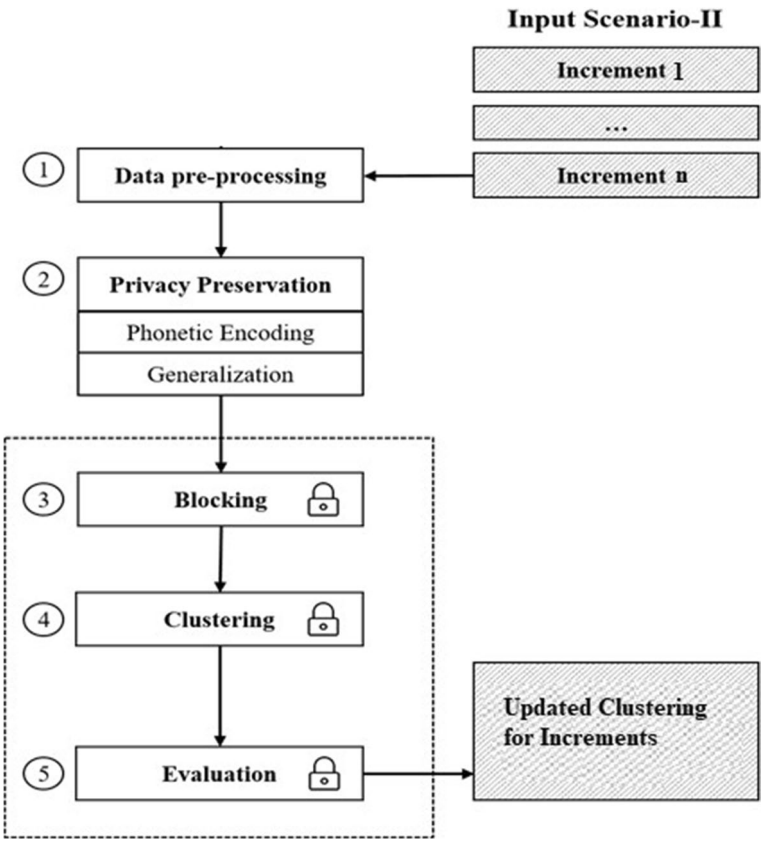

Fig. 6 PPIRL, an end-to-end framework steps for increments

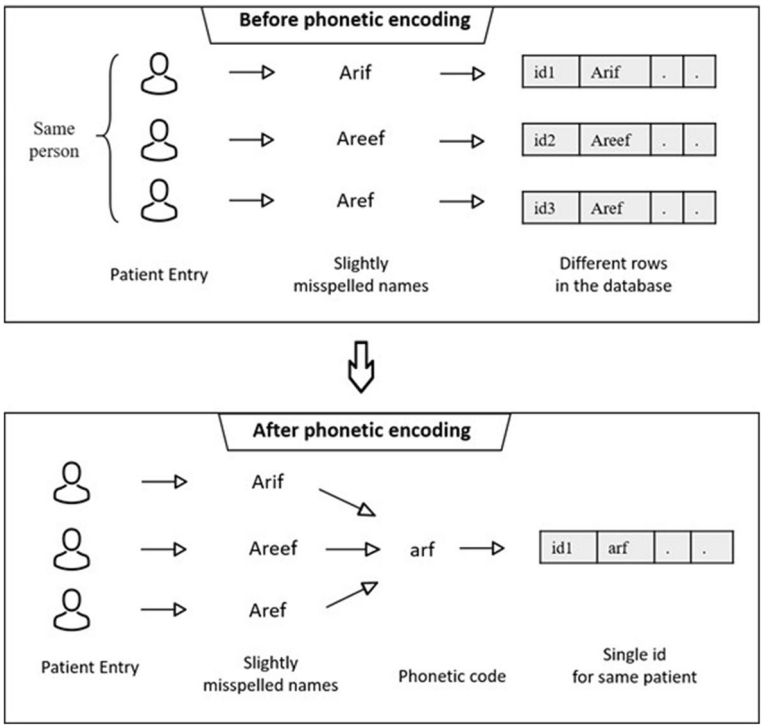

Fig. 7 Process of phonetic encoding

We used nameSignificance algorithm for phonetic encoding as it produces better results than Soundex, Metaphone, and some other commonly used phonetic algorithms.

K-anonymization method

K-anonymization is a popular generalization algorithm. The main purpose of generalization is to help overcome the problem that lies with record linkage which is the re-identification of entities. The data generalization process generalizes data in a way that re-identifying the data to its source record is quite impossible. There are many generalization techniques. Among them, the K-anonymization method has been proven to be an effective privacy technique that can preserve the privacy of record linkage results [5]. In the K-anonymization technique, we assume that data related to a specific person is gathered in a dataset. The anonymization process is started

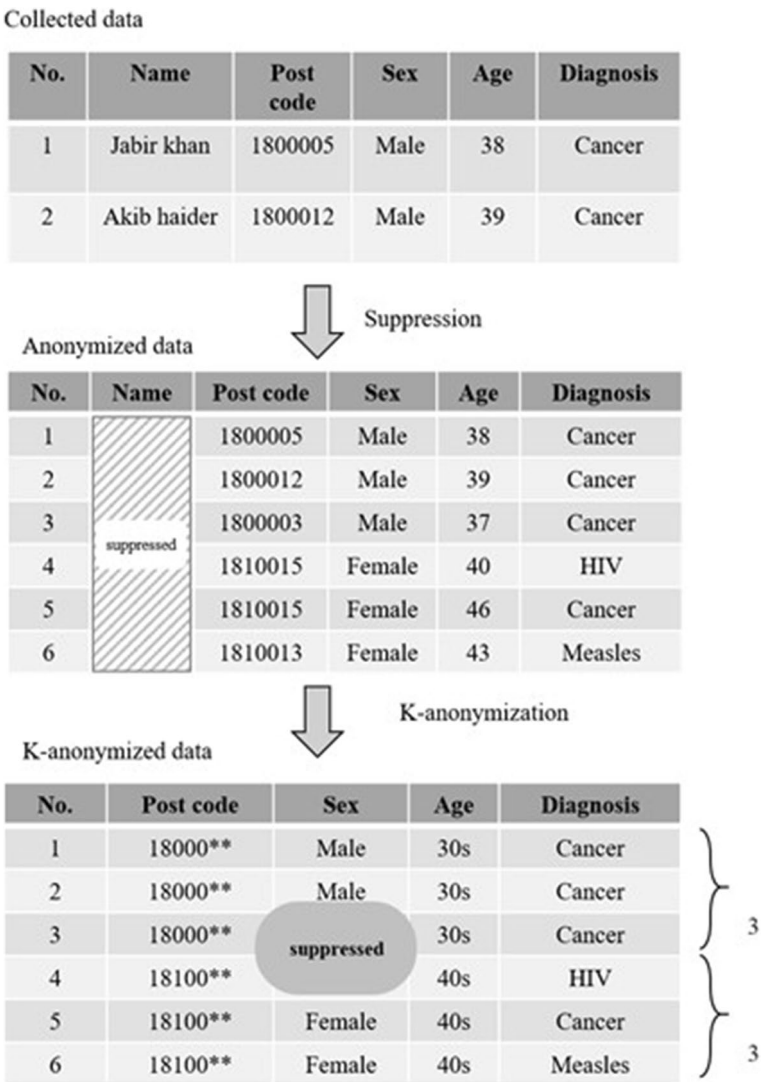

Fig. 8 Illustration of generalization algorithm

by removing all the identifying features like SSN, explicit identifiers, etc. Even after removing the identifying features, it is possible to find a person's data by finding a pattern in other features. To tackle that K-anonymization generalizes the feature values as much as possible with a value of K either being fixed at the beginning of getting an adaptive value as the linkage process continues. Fig. 7 explains the K-anonymization process elaborately.

In Fig. 8, we can see that common identifying feature such as name is suppressed at the very beginning. Then, other features which could be used to identify a person in a dataset are generalized by varying the value of K at a different time. In our case, we have used the adaptive value of K which allows us to select the best value of K depending on the results found.

Blocking

Before moving on to the explanation of blocking methods for our framework we should first understand the due importance of blocking as an inseparable phase of new incremental record linkage or standard record linkage procedures [4, 37]. To find the best match of records record pair comparison is needed to be done on the dataset. When the possible quantity of matched records increases the number of record pair comparisons also increase rapidly. For large datasets, this approach is computationally impractical. Blocking allows us to divide the whole dataset into blocks depending on some criteria. There are many existing techniques to achieve the task of blocking.

Feature set for blocking

For our framework, we have five identifiable attributes of the patient which we have applied to our clustering algorithm. These attributes are Patient Name, Gender, Age, Contact Number, and Address. We have to select an attribute or a group of attributes as a blocking key such that it maintains a balance between the computation and communication cost. We have carried out several experiments to find the best set of features which would bring effective results. To achieve this, we took all the possible feature sets from the five already selected features for our framework, and after careful inspection of the results, we selected the one with the best outcome. In our case, the Gender-Address set of features was better than other options. So, for further experiments, we have used this set of features for blocking.

Clustering

There are many clustering algorithms available. Because of the simple approach and easy implementation process compared to existing approaches we have opted to go for agglomerative hierarchical clustering.

The Davies-Bouldin index (DB) is calculated as follows. For each cluster C, the similarities between C and all other clusters are computed, and the highest value is assigned to C as its cluster similarity. Then the DB index can be obtained by averaging all the cluster similarities. The smaller the index is, the better the clustering result is. By minimizing this index, clusters are the most distinct from each other and,

therefore, achieve the best partition. The Davies-Bouldin index was originally defined for a Euclidean space; applying it to record linkage requires some adjustment for the definition of distance. We adopt the definition, described as follows. For each cluster C, the intra-cluster distance is defined as the complement of average similarity between records in the cluster; that is,

$$D(C) = 1 - Avg_{r,r' \in C} sim(r, r')$$

For each pair of distinct clusters C and C', the inter-cluster distance is defined as the complement of average similarity between records across the clusters; that is,

$$D(C, C') = 1 - Avg_{r \in C, r' \in C'} sim(r, r')$$

The separation measure between C and C'is then defined as

$$M(C, C') = \frac{D(C) + D(C') + \alpha}{D(C, C') + \beta}$$

Where alpha and beta are small positive numbers such that the denominator or numerator would affect the result even when the other is 0.2 for each cluster C, we define its separation measure as

$$M(C) = max_{C' \neq C} M(C, C')$$

DB-index is defined as the average separation measure for all clusters and we wish to minimize it.

Correlation penalty: For each pair of records in the same cluster, there is a cohesion penalty being the complement of the similarity; for each pair of records in different clusters, there is a correlation penalty being the similarity. We wish to minimize the sum of the penalties.

$$CC(L_G) = \sum_{C \in L_G, r, r' \in C} (1 - sim(r, r')) + \sum_{C, C' \in L_G, C \neq C', r \in C, r' \in C'} sim(r, r')$$

A special case for correlation clustering is when we take binary similarities: the similarity between two records is either 0 (dissimilar) or 1 (similar).

We have the group averaged agglomerative version for our framework. Although ward's criterion is popularly used to compute the distance between two clusters during agglomerative clustering in our case we needed something customized which would serve our purpose directly. Ward's criterion uses the K-means squared error criterion to determine the distance. It is also interpreted as the squared Euclidean distance between the centroids of the merged clusters. However, in our clustering, we did not use centroids rather all the data objects' average similarity was used to determine the suitable clusters. In order to achieve that we applied our similarity metric. There we examined the identifying attributes of a patient and calculated similarity with pre-assigned weights of the attributes. We chose the weights from domain knowledge, global standards, and local trends.

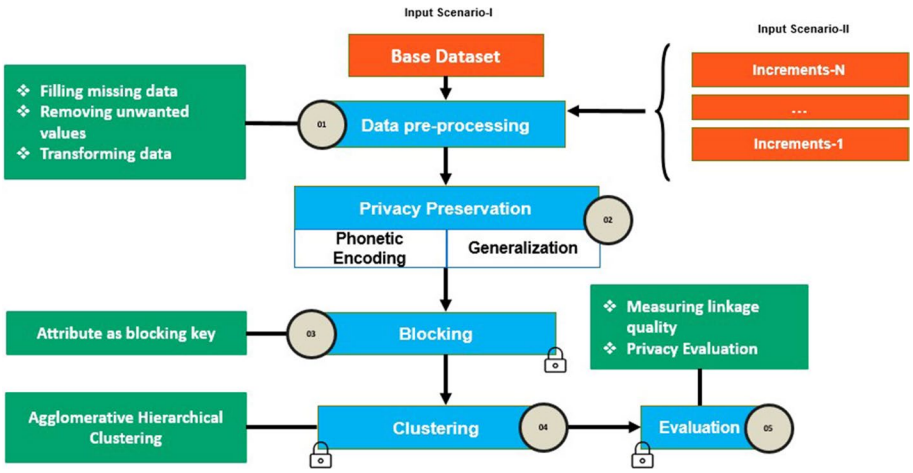

Fig. 9 Impact of incremental feature on the PPiRL framework

Impact of incremental approach in PPRL

Figures 5 and 6 provide a comparison of the impact of the incremental feature on the overall framework. Figure 5 represents the part of the framework at the initial state with a base dataset. That is the same as the traditional batch processing approach. Here the whole dataset is clustered using step 3, 4 and 5 that is based on blocking, clustering and evaluation results. But when the increments come, the framework acts differently than the batch processing framework. It does only process the increment part of the dataset and adjusts it with the previous clustering results, shown in Fig. 6. The incremental record linkage (IRL) process preserves the clusters developed from the base dataset D and merges the records from the incremental dataset ΔD using any similarity function. Figure 4 illustrates the inner concept of the IRL Framework. Figure 9 illustrates the PPiRL framework for clustering of a base dataset as well as multiple increments.

IRL improves the linkage speed significantly compared to its corresponding batch linkage algorithm especially if the increments are small. Specifically, the computation of f(D, ΔD, ρ_D) should be faster than the computation of F(D + ΔD) if $|\Delta D| << |D|$ holds. For example, if the Base dataset contains 1 Million records and increment 1 contains 10 thousand records, in the batch processing framework, it will need to process 1.01 million records to produce the correct clusters including the 1st increment. On the other hand, our proposed incremental framework will only need to process 10 thousand records instead of 1.01 million records to produce correct clusters incorporating the 1st increment. So our proposed framework can produce record linkage much faster with producing similar results as batch processing which is present in the "Result" section.

Evaluation

The outcome of the PPiRL technique needs to be evaluated from different aspects. As our main goal is to combine the privacy technique with incremental record linkage, so we need to evaluate our framework in light of privacy and clustering validation.

Linkage evaluation

Evaluation of linkage evaluates the quality of clustering results. Attaining success in different clustering applications has been acknowledged as a key task. Linkage evaluation can be implemented in two ways. External validation of clustering can be implemented when external information like the class label of a cluster is already present for the dataset. However, when this type of external validating information is not present internal validation measures could be used. For external validation, we have used F-measure to validate the outcomes of our framework. For internal validation, we have used the Davies Bouldin Index penalty and correlation penalty.

Privacy evaluation

To strengthen the privacy of our framework, it is imperative to evaluate the framework with more than one type of privacy analysis technique. Hence the implementation of frequency analysis and dictionary attack analysis was integrated with the framework. To ensure the privacy aspect of our framework we have gone for several privacy evaluation techniques. Frequency analysis, dictionary attack, and adversary model simulation proof testing are the key evaluation measures that we have taken for the framework.

Experimental results

In this section, we present the results of different experiments based on real-world and synthetic datasets. Our algorithm is presented next.

Algorithm 1: PPiRL

```
Input: Recordset for Record Linkage (Base dataset or an Increment);
Output: Record Linkage results;
while !End of Recordset do
    Preprocess recordset;
    Feature Selection;
    Standardization;
    Phonetic Encoding of Text attributes;
    K-Anonymization of Numeric attributes;
    Apply clustering;
end
```

Data pre-processing

We experimented with a real-world dataset. The dataset contains 65,000 records of Bangladeshi patients from different healthcare organizations. We randomly divide 50,000 patient records as our base dataset. We divided the remaining 15,000 patient-records into three increment datasets ΔD_1, ΔD_2, and ΔD_3. Each of them contains 5,000 records.

Feature selection

Feature selection is a procedure where a subset of original features is selected by following some criteria. To select necessary features for our framework, we have followed the basic three steps of feature selection. First, we have collected all features from the record set to be linked. Second, we have generated a candidate set, a subset of the whole set, which contained some selected features from the dataset using domain knowledge. Then, we evaluated the clustering result with the candidate set. Finally, after several

Table 1 Feature Selection

Features in the raw dataset	Selected identifiable features
Invoice No	Patient Name
Invoice Date	Gender
Patient Name	Age
Gender	Contact Number
AGE	Address
Contact Number	
Address	
Test Name	
Delivery Date	
Department	
Sample	
Test Attribute	
Result	
Unit	
Reference Value	

Table 2 Age normalization

Age	Actual Details	Scaled Age
7 Y 4 M	7 years 4 months	7 years
11 Y 6 M	11 years 6 months	12 years
3 D	3 days	1 year
9 M	9 months	1 year

iterations, we have found a set of five features. They are the patient's name, gender, age, contact number, and address. Table 1 lists the selected features via this process. In the left column of Table 1, we can see fifteen features that are fundamental in the raw dataset, and in the right column, we can see the finally selected five features. Details of feature selection are out of the scope of this paper.

Normalization of age values

Data standardization plays a key role to ensure the quality of data. If data mining lacks proper standardization of data, it results in bad data which creates a multitude of negative effects. As our framework deals with sensitive healthcare data, it becomes an obvious necessity. One of the features in the healthcare dataset is the age of patients. The framework will produce a bad result if this feature is not dealt with properly. Because most of the time this feature is recorded with different types of units and sometimes a mixture of units. For neonates, in their early years' healthcare organizations tend to use days and months for keeping track of the babies' age. For adults, although days are hardly used months are used repeatedly. So, having a uniform age unit is needed for data mining tasks. We applied normalization techniques to transform the age to hold only age in year format.

In Table 2, we can see the actual age values that appear in the raw dataset such as 7 Y 4 M. This type of changing values in units is harmful to our calculation. So, we have transformed the age values to a standard which can be seen in the right-most column of Table 2. We can see the transformed age values as they all appear with the year as their unit of representation. This helps the calculation of the Age feature in the patient dataset.

Address standardization

The addresses provided by the patients in healthcare data are also very noisy and unstructured. For that reason, we have come up with the idea of extracting the address in a strict format and following the standard provided by the Bangladesh Bureau of Statistics (BBS) of the Ministry of Planning. BBS provides a GEO code list up to the Upazila label of Bangladesh. So, we extracted the raw addresses and formatted them in the desired format of BSS. Then we applied the GEO codes from the code list with the help of our algorithm. Below we can see both the extracted addresses and the mapped geocodes for the corresponding addresses. In Table 3, the geocodes are shown as they are formed.

Firstly, we transform the addresses of each patient into a usable general form with the order where Upzilla (Sub-district), Zilla(District), and Division are maintained. Then from this, we generate the geocoded mapping for each address.

Experimental setup

Working environment: The machine used for the experiment was a Windows machine running a 64-bit Windows 10 operating system on an Intel®Core™ i7-5500U CPU @ 2.40 GHz. The machine has an 8.00 GB RAM capacity and 1 TB hard disk space. Implementations: To determine the effectiveness of our framework, we implemented the following algorithms:

– nameGist, the phonetic encoding algorithm, groups the similar-sounding names together and gives privacy to the 'Name' feature as well.

Table 3 Mapping address to geocode

Extracted Address	Mapped Geocode
Anowara, Chittagong, Chittagong	201504
Saturia, Manikganj, Dhaka	305670
Anowara, Chittagong, Chittagong	201504
Patenga, Chittagong, Chittagong	201565
Fakirhat, Bagerhat, Khulna	400134
Rampal, Bagerhat, Khulna	400173
Birampur, Dinajpur, Rangpur	552710
Barlekha, Maulvi Bazar, Rangpur	605814
Adamdighi, Bogra, Rajshahi	501006
Companiganj, Sylhet, Sylhet	609127

Table 4 Blocking results using address/geocode as the blocking key

Clustering process	Correlation Clustering		Naïve clustering	
Noise	F-measure	time(s)	F-measure	Time(s)
0%	94.21%	4.06	94.10%	2.54
5%	91.30%	6.41	92.20%	3.85
10%	89.40%	7.12	89.80%	5.2

Table 5 Penalty evaluation for naive and correlation clustering

Clustering	Evaluation measure	Penalty
Naïve Clustering	Correlation Penalty	116.92
	DB Index Penalty	71.96
Correlation clustering	Correlation Penalty	115.02
	Correlation Penalty	52.12

- K-Anonymization, the privacy-preserving algorithm, ensures the generalization of 'Contact,' 'Address' features.
- NAIVE, the incremental baseline algorithm, compares each inserted record with existing clusters, then either adds it into an existing cluster or creates a new cluster for it.
- Correlation Clustering applies a correlation penalty to get the best cluster results while implementing clustering.

Linkage evaluation

External validation measure results

We have measured the efficiency, quality, and privacy of our algorithms. For efficiency, we considered execution time. We repeated the experiments 100 times and reported the average execution time. As we focused on clustering, we only reported clustering time. For quality, we report (1) the penalty (i.e., cut inter-cluster and missing intra-cluster edges) and (2) the F-measure if we have the gold standard. Here, precision measures among the pairs of records that are clustered together, how many are correct; recall measures among the pairs of records that refer to the same real-world entity, how many are clustered together, and the F-measure is computed as:

$$F = \frac{2 * Precision * Recall}{Precision + Recall}$$

For privacy, we considered the frequency distribution of various sensitive features in the dataset. We have applied two types of clustering algorithms for our incremental record linkage application. One of the algorithms is the Naïve incremental algorithm, and the other one is the correlation clustering approach. We applied the algorithms to our dataset with varying noise. We introduced intentional noise in our dataset ranging from 5% to 10% of the total records in the dataset. The results can be understood in two aspects, accuracy, and the other time efficiency. Both these two aspects give us the overall

performance of a blocking key. We have used Geocode as the blocking key. As we have a large dataset, little improvement in performance means a lot for the overall result of the algorithm. When the Upazilla part of the Geocode is used as the blocking key, our algorithm divides the full dataset into as many parts as the variants of Upazilla in an address appear because there are many Upazillas all around the country. So, the accuracy of the algorithm improves significantly at the same time the number of records in a block also reduces to a balanced level as the dataset grows. So far, this blocking behaves better than the other blocking approaches. In Table 5 we can see the performance of Naïve clustering for only Gender-based blocking.

Table 4 clearly shows the clustering results for Naïve clustering for linkage purposes. It gets the best threshold value from iteratively going through all the threshold values. Then for various noise percentages of the dataset, it calculates precision, recall, and F-measure of the linkage results.

Then we moved on to calculate the correlation clustering results as we also implemented it in order to compare it with an existing solution. Table 5 indicates the results we received for correlation clustering using Gender as the only blocking key. In Table 5 we can see the result for several types of noise values and different results for correlation clustering. As Correlation clustering checks all the pairs between two comparing clusters iteratively we can see that even with no noise in the dataset it takes the most time.

Internal validation measure results

To evaluate the linkage results we have also resorted to experiments related to internal evaluation matrices. We have used correlation penalty and DB-Index penalty particularly to evaluate the linkage quality. The lesser the penalty, the better the results. In Table 6, we have shown the correlation penalty and DB-Index penalty for Naïve clustering. The equation for penalty is presented earlier in the "Clustering" sub-section.

In Table 5, we can see that the Correlation penalty is much higher than the DB-Index penalty for Naïve clustering. As DB-Index has a more robust formula, we can say the performance of our algorithm is better in this regard. We have also shown the correlation penalty and DB-Index penalty for Correlation clustering for linkage purposes on our dataset.

Privacy evaluation

For evaluation of the privacy preservation of PPiRL, we have used three widely used techniques: a) Frequency analysis, b) Information gain, and c) Dictionary attack. These techniques are discussed in the following sub-sections.

Frequency analysis

The frequency distribution of the characters of certain attributes in a dataset may cause information exposure. In our experimental dataset, there are several identifying features of a patient. Among them, the 'Name' feature is a sensitive one. To get a hold of this feature one way is to get the frequency distribution of English letters occurring in the names. If the frequency of letters remains the same even after applying privacy techniques to encode the names, then the names can be in the hands of an unwanted outsider or in the worst case a hacker using frequency distribution. Figure 5 is the representation

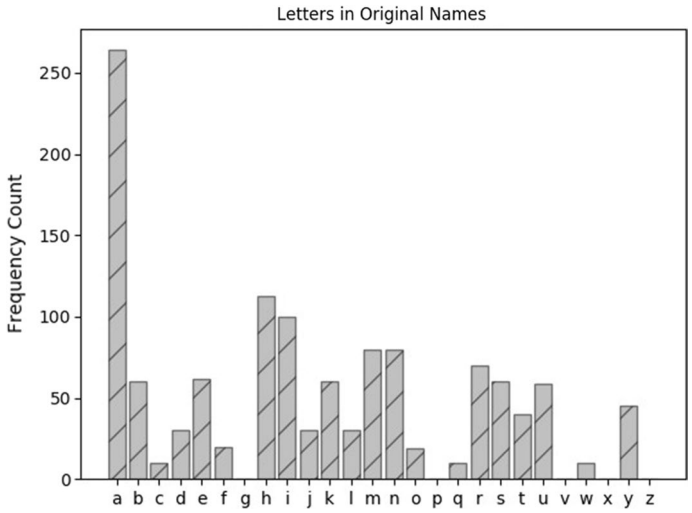

Fig. 10 Frequency analysis of original 'Name' attribute

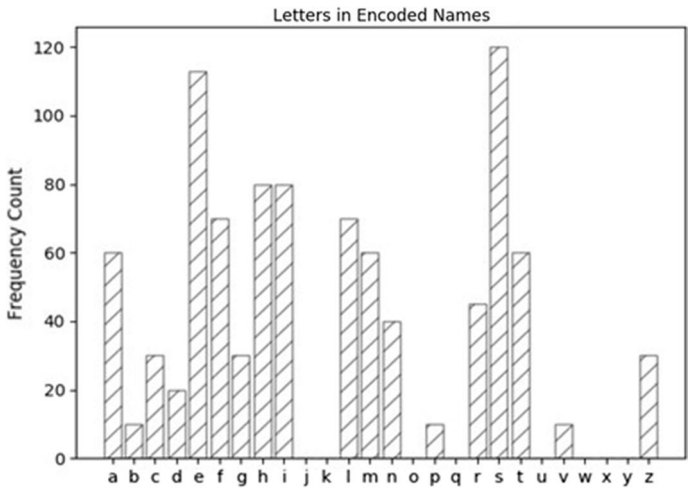

Fig. 11 Frequency analysis of 'Name' attribute after privacy enforcement

of the frequency of letters found in the original names before any privacy algorithm is applied.

In Fig. 10, we can see the frequency of each letter found in the Name feature of the patient dataset. The figure is a histogram representation of the frequency of the letters. It can be deducted from the graph that certain letters appear more than others making the graph a skewed graph. To check the authenticity of the privacy algorithm that works on the Name feature we have a type of phonetic encoding algorithm named nameGist which encodes the name of the patient into a code depending on the phonetic characteristics of the name. We have illustrated the frequency of letters in the names of the patient after the application of this nameGist algorithm.

In Fig. 11, we can see the frequency of each letter found in the encoded version of the Name feature of the patient dataset. The figure is a graph representation of the frequency

of the letters. This graph is significantly different from the graph that is shown in Fig. 10 because there was only one letter that had a frequency higher than 100, but in this graph, we can see that more than two letters exceed the label of 100. Also, after more close inspection we can say that the frequency of this graph is different from the last one and has a more well-balanced frequency of the letters. This balance is crucial as the hackers rely on the similarity of frequency to steal valuable information from health datasets. The difference in frequency count is visible. It is also notable that the frequency count has decreased significantly in the second figure which makes the privacy of the Name attribute of the patients quite safe. Thus, if a hacker even gets hold of the encrypted dataset, it is quite impossible to get the actual names of the patients from it.

Dictionary attack

A dictionary attack is usually carried out to break passwords or similarly encrypted data in a database with the help of an available digital dictionary [31]. To carry out a successful dictionary attack, a hacker must have access to a dictionary or list of frequently used words or vocabularies. General dictionaries are useful in this matter as they provide millions of words that could be used to create a password for a user. Our dataset does not contain passwords, but it contains encoded privacy names.

As a dictionary contains so many words that normally people's usable passwords contain in there and hackers used a technique to go through all the words in a dictionary and find a word or something to encrypt the data. Our encoded data have been checked by a security research team to evaluate whether they can survive the dictionary attack with the help of the THC Hydra tool [20]. They found that our key is not a very frequent dictionary word that's why our key is marked as safe from a dictionary attack.

Information gain

By simulating the framework under different adversary models, we can evaluate the proof of privacy of PPiRL. The less information is extracted from the framework, the safer it is. Here we have considered the popular Honest-but-curious (HBC) Adversary Model. In the HBC model, each of the parties is obliged to follow the protocol. Here, a party does not forget the knowledge that it learns during the information exchange. In other words, all parties are curious, in the sense that they try to find out information about other parties, as much as possible, despite following the protocol [34]. In the perspective of HBC, a protocol is secure if and only if all parties involved have gained no new knowledge at the end of the exchange other than what they would have learned from the output of the record pairs classified as matches. Most of the PPRL solutions proposed in the literature assume the HBC adversary model [40].

We have evaluated the privacy preservation of our proposed technique using information gain. For information gain, first, we have to calculate the entropy and conditional entropy of a message. Entropy is a measure of the total information of a message. It is a probability distribution function overall set of possible messages. The equation for entropy is given below:

Table 6 Entropy of individual attributes and concatenated data

Attribute	Entropy(bit/Character)
Name	3.2
Gender+birthrange	2.25
Address	3.32
Concatenate value	4.25

$$H(X) = -\sum_{j=1}^{m} p_j log p_j$$

Low entropy means low uncertainty as a result of higher predictability. In Table 6 calculated value of entropy for the sensitive attributes and concatenated value is presented. Here we calculated entropy for a bit per character in a message. We can see from the table that concatenated attribute's entropy is higher than that of individual attribute thus producing lower predictability of real data.

Conditional entropy is another function for evaluating information gain. It measures the amount of uncertainty in predicting the value of random variable Y given X. The equation is given below:

$$H(Y|X) = \sum_{j} p(X = v) H(Y|X = v)$$

Information gain is a metric for measuring the difficulty of revealing variable Y given X. The formula is given below:

$$IG(Y|X) = H(Y)H(Y|X)$$

So in our output data, there are other features like disease, the result of diagnosis, or other features available as we provide privacy to only personal sensitive information. So in HBC settings, if a party is curious about sensitive features and tries to reveal the data, it has to use these features.

In Table 7 we have presented a calculation of information gained for PPiRL. As different words have different lengths so we measured the percentage of the gain. From the percentage, we can clearly see that only 19% information can be understood by a party which is very less.

If an exchanging party got both the plain text and the privacy preserved text from our framework, the party can only reveal 19.91% information in the HBC adversary model. The above statement is for the case of PPiRL. But for IRL, the information gain is near about 100%.

Table 7 Information gain

Adversary model name	Information gain
Honest-but-curious behavior(HBC)	19.91%

Table 8 Comparison between PPiRL and batch record linkage

Clustering	PPiRL		Batch	
			Record Linkage	
Update	F-measure	Time(s)	F-measure	Time(s)
Initial	94.21%	2.54	95.70%	10.33
Increment-I	91.30%	3.85	93.20%	14.5
Increment-II	89.40%	5.2	91.40%	16.2

Table 9 Comparison between PPiRL and IRL

Feature	Framework	
	IRL	PPIRL
Privacy Preservation Technique	None	Phonetic encoding & Generalization
Information gain by other party	Full	19%
Linkage quality	95%	91%

Comparison of PPiRL with batch-PPRL

The traditional or batch privacy preserved record linkage (PPRL) process considers all the new records and does not maintain the cluster from the last linkage when they arrive [39]. The process of PPRL can be understood in Fig. 2. On the other side, PPiRL keeps track of the cluster of the last linkage. So, when new updates arrive, it performs faster than Batch linkage. The difference between the PPiRL framework with traditional PPRL can be understood in Fig. 9. As our proposed PPiRL framework adopted the incremental approach, its record linkage speed is much faster than the traditional PPRL. Table 8 briefly compares the performance of these two approaches.

In Table 8, the linkage is tested as per the datasets arrive at the algorithms. There is an initial dataset that has the base records and clusters. Increment-I and Increment-II arrive with approximately 5,000 records. For these updates, PPiRL takes much less time than Batch. Although the result in the batch is slightly better than PPiRL as Batch is an exhaustive process. But when the base dataset becomes larger and increments are much smaller comparing the base dataset, the efficiency of PPiRL is much better than batch linkage.

Comparison of PPiRL with incremental record linkage

Traditional Incremental Record Linkage (IRL) approaches perform record linkage in an incremental fashion to improve performance but do not support privacy preservation [18]. That is IRL's focus on performance, not on privacy. We have also compared PPiRL with IRL. Privacy preservation techniques are not used in IRL so we have an extra advantage in PPiRL that we can see in Table 9. We have gotten privacy assurance of 81% in our PPiRL framework. In IRL, using Honest-but-Curious (HBC) model, the other party or intruder may gain the full details of the exchanging entities of the records to be linked. But in our approach, a maximum of 19% of information can be uncovered by the other party or intruder. So our proposed PPiRL framework has a

clear advantage over the traditional IRL framework when the privacy of the records is a concern, i.e. medical or financial records. Although our framework got a little less linkage quality, it was a tradeoff due to privacy techniques which are considered in this type of sensitive research.

Discussion

Our proposed technique Privacy Preserved Incremental Record Linkage (PPiRL) is a novel concept that incorporates the good features of both Privacy Preserved Record Linkage (PPRL) and Incremental Record Linkage (IRL). The key benefit of IRL is its faster processing time than traditional batch record linkage as IRL process only the increment dataset rather than the whole dataset. As in PPiRL, we have adopted an incremental approach rather than a batch processing approach, it takes much less time compared with the traditional PPRL techniques. For clustering the initial dataset plus two increments PPiRL takes 11.59 seconds which is much smaller than the benchmark PPRL technique which takes 41.03 seconds. So our framework performs linkage around four times faster as it uses the incremental approach rather than the batch processing approach to process new data. With time, when a frequently smaller chunk of data will reach for record linkage, the performance of PPiRL will be much better. For the linkage of sensitive data e.g., health, finances etc. our proposed PPiRL has a big advancement over the benchmark IRL techniques, that is PPiRL ensures privacy of the records used for linkage and IRL failed to provide any privacy. It is apparent from Table 9. As for preserving the privacy of the sensitive data, we have used multiple algorithms in PPiRL, so its linkage accuracy is a little bit lower than IRL as IRL does not use any privacy preservation. We think this little drop in accuracy is quite acceptable for the application for whom PPiRL is intended to develop.

Summary

This paper proposes an end-to-end framework that conducts privacy-preserving algorithms as well as record linkage algorithms in an incremental fashion. Our algorithms ensure the privacy of the sensitive records and also maintain the linkage of the records by creating a proper cluster of similar records. Being the first to experiment in this field, we could only apply very few algorithms to test our framework. Combining privacy with incremental record linkage has paved the way to secure linkage of sensitive data residing in several public and private organizations meeting the demand of the big data era. Experimental results with 65000 records from multiple datasets show that our framework can achieve around 90% correct record linkage with much reduced times. Different privacy attacks, executed by an external body, showed that our framework is stable against well-known attacks e.g., dictionary attacks, and frequency attacks. The information gained from the exchange record using the HBC model is also less than 20%. In the future, we want to improve the linkage quality, privacy, and performance of the framework using other state-of-the-art algorithms.

Abbreviations

PPRL Privacy-Preserving Record Linkage

IRL	Incremental Record Linkage
PPiRL	Privacy-preserving incremental record linkage
PHI	Protected Health Information
SSN	Social security number
kNN	k-Nearest Neighbor
SVM	Support Vector Machine
CPP	Clique Partition Problem
RMSE	Root Mean Square Error
HBC	Honest-but-curious
BBS	Bangladesh Bureau of Statistics
BUET	Bangladesh University of Engineering and Technology

Acknowledgments
The authors thankfully acknowledge the support of the members of the IIUC Data Science Research Group (IDSRG) of International Islamic University Chittagong (IIUC), especially Mohammad Sheikh Ghazanfar, Md. Shahnewaz Refath, and Mohammad Safiul Basher Tarek. A significant portion of this research is performed in the Graduate Complex, Dept. of CSE, Bangladesh University of Engineering and Technology (BUET).

Author contributions
Khan SI has a major contribution to the idea of research and in designing the experiments. He also developed the algorithms used in the research. Khan ABA also contributed to the concept and supported writing the manuscript and drawing the figures. Hoque ASM formulates the problem and writes the definitions. He also proofreads the manuscript. All authors read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declaration

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 19 January 2022 Accepted: 20 October 2022
Published online: 14 November 2022

References

1. Al-Lawati A, Lee D, McDaniel P. Blocking-aware private record linkage. In: Proceedings of the 2nd international workshop on Information quality in information systems. ACM; 2005. p. 59–68
2. Bachteler T, Schnell R, Reiher J. An empirical comparison of approaches to approximate string matching in private record linkage. In: Proceedings of statistics canada symposium, vol. 2010. Citeseer; 2010
3. Batini C, Scannapieco M, et al. Data and information quality. Cham: Springer International Publishing. Google Scholar; 2016.
4. Baxter R, Christen P, Churches T, et al. A comparison of fast blocking methods for record linkage. In: ACM SIGKDD, vol. 3. Citeseer; 2003. p. 25–27
5. Bayardo RJ, Agrawal R. Data privacy through optimal k-anonymization. In: 21st International conference on data engineering (ICDE'05). IEEE; 2005. p. 217–228
6. Bellahsène Z, Bonifati A, Rahm E. Schema matching and mapping. Springer; 2011.
7. Bleiholder J, Naumann F. Data fusion. ACM Comput Surv (CSUR). 2009;41(1):1.
8. Charikar M, Chekuri C, Feder T, Motwani R. Incremental clustering and dynamic information retrieval. SIAM J Comput. 2004;33(6):1417–40.
9. Christen P. A comparison of personal name matching: Techniques and practical issues. In: Sixth IEEE international conference on Data mining workshops, 2006. ICDM Workshops 2006. IEEE; 2006. p. 290–294
10. Christen P. Development and user experiences of an open source data cleaning, deduplication and record linkage system. ACM SIGKDD Explorations Newsl. 2009;11(1):39–48.
11. Christen P. A survey of indexing techniques for scalable record linkage and deduplication. IEEE Trans Knowl Data Eng. 2012;24(9):1537–55.
12. Christen P, Goiser K. Quality and complexity measures for data linkage and deduplication. In: Quality measures in data mining. Springer; 2007. p. 127–151
13. Churches T, Christen P. Some methods for blindfolded record linkage. BMC Med Inform Decis Mak. 2004;4(1):9.

14. Cohen WW, Richman J. Learning to match and cluster large high-dimensional data sets for data integration. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM; 2002. p. 475–480
15. Elmagarmid AK, Ipeirotis PG, Verykios VS. Duplicate record detection: a survey. IEEE Trans Knowl Data Eng. 2006;19(1):1–16.
16. Fellegi IP, Sunter AB. A theory for record linkage. J Am Stat Assoc. 1969;64(328):1183–210.
17. Franzak F, Pitta D, Fritsche S. Online relationships and the consumer's right to privacy. J Consum Mark. 2001;18(7):631–42.
18. Gruenheid A, Dong XL, Srivastava D. Incremental record linkage. Proc VLDB Endow. 2014;7(9):697–708.
19. Gu L, Baxter R. Decision models for record linkage. In: Data mining. Springer; 2006. p. 146–160
20. Hauser V. The hacker's choice, a very fast network logon cracker which support many different services. 2010
21. Hernández MA, Stolfo SJ. Real-world data is dirty: data cleansing and the merge/purge problem. Data Min Knowl Disc. 1998;2(1):9–37.
22. Herzog TN, Scheuren FJ, Winkler WE. Data quality and record linkage techniques. Springer Science & Business Media; 2007.
23. Humer C, Finkle J. Your medical record is worth more to hackers than your credit card. Reuters.com US Edition 24 (2014)
24. Inan A, Kantarcioglu M, Bertino E, Scannapieco M. A hybrid approach to private record linkage. In: IEEE 24th international conference on data engineering, 2008. ICDE 2008. IEEE; 2008. p. 496–505
25. Khan SI, Hoque ASL. Privacy and security problems of national health data warehouse: a convenient solution for developing countries. In: 2016 international conference on networking systems and security (NSysS). IEEE; 2016. p. 1–6
26. Khan SI, Hoque ASML. An analysis of the problems for health data integration in Bangladesh. In: 2016 International conference on innovations in science, engineering and technology (ICISET). IEEE; 2016. p. 1–4
27. Khan SI, Latiful Hoque ASM. Digital health data: a comprehensive review of privacy and security risks and some recommendations. Comput Sci J Mold 2016; 24(2)
28. Lee YW, Pipino LL, Funk JD, Wang RY. Journey to data quality. The MIT Press; 2009.
29. Mathieu C, Sankur O, Schudy W. Online correlation clustering. 2010. arXiv preprint arXiv:1001.0920
30. Mukherjee A, Nath P. A model of trust in online relationship banking. I J Bank Market. 2003;21(1):5–15.
31. Narayanan A, Shmatikov V. Fast dictionary attacks on passwords using time-space tradeoff. In: Proceedings of the 12th ACM conference on computer and communications security. 2005. p. 364–372
32. Do Nascimento DC, Pires CES, Mestre DG. Heuristic-based approaches for speeding up incremental record linkage. J Syst Softw. 2018;137:335–54.
33. Naumann F, Herschel M. An introduction to duplicate detection. Synth Lect Data Manage. 2010;2(1):1–87.
34. Paverd A, Martin A, Brown I. Modelling and automatically analysing privacy properties for honest-but-curious adversaries. Tech Rep. 2014
35. Rahm E, Do HH. Data cleaning: problems and current approaches. IEEE Data Eng Bull. 2000;23(4):3–13.
36. Sherstobitoff R. Anatomy of a data breach. Inf Security J Glob Perspect. 2008;17(5–6):247–52.
37. Steorts RC, Ventura SL, Sadinle M, Fienberg SE. A comparison of blocking methods for record linkage. In: International conference on privacy in statistical databases. Springer; 2014. p. 253–268
38. Tauer G, Date K, Nagi R, Sudit M. An incremental graph-partitioning algorithm for entity resolution. Inf Fusion. 2019;46:171–83.
39. Vatsalan D, Christen P. Scalable privacy-preserving record linkage for multiple databases. In: Proceedings of the 23rd ACM international conference on conference on information and knowledge management. 2014. p. 1795–1798
40. Vatsalan D, Christen P, Verykios VS. A taxonomy of privacy-preserving record linkage techniques. Inf Syst. 2013;38(6):946–69.
41. Vatsalan D, Sehili Z, Christen P, Rahm E. Privacy-preserving record linkage for big data: current approaches and research challenges. In: Handbook of big data technologies. Springer; 2017. p. 851–895
42. Verykios VS, Karakasidis A, Mitrogiannis VK. Privacy preserving record linkage approaches. Int J Data Min Model Manage. 2009;1(2):206–21.
43. Weber SC, Lowe H, Das A, Ferris T. A simple heuristic for blindfolded record linkage. J Am Med Inform Assoc. 2012;19(e1):e157–61.
44. Whang SE, Garcia-Molina H. Entity resolution with evolving rules. Proc VLDB Endow. 2010;3(1–2):1326–37.
45. Whang SE, Garcia-Molina H. Incremental entity resolution on rules and data. VLDB J Int J Very Large Data Bases. 2014;23(1):77–102.
46. Yakout M, Atallah MJ, Elmagarmid A. Efficient private record linkage. In: IEEE 25th international conference on data engineering, 2009. ICDE'09. IEEE; 2009. p. 1283–1286

Publisher's Note