Information Systems 38 (2013) 946–969

Contents lists available at SciVerse ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

# A taxonomy of privacy-preserving record linkage techniques

Dinusha Vatsalan [a,*], Peter Christen [a], Vassilios S. Verykios [b]

[a] Research School of Computer Science, The Australian National University, Canberra, ACT 0200, Australia
[b] School of Science and Technology, Hellenic Open University, Patras, Greece

ARTICLE INFO

Available online 28 November 2012

*Keywords:*
Record linkage
Data matching
Entity resolution
Data quality
Privacy techniques
Survey

ABSTRACT

The process of identifying which records in two or more databases correspond to the same entity is an important aspect of data quality activities such as data pre-processing and data integration. Known as record linkage, data matching or entity resolution, this process has attracted interest from researchers in fields such as databases and data warehousing, data mining, information systems, and machine learning. Record linkage has various challenges, including scalability to large databases, accurate matching and classification, and privacy and confidentiality. The latter challenge arises because commonly personal identifying data, such as names, addresses and dates of birth of individuals, are used in the linkage process. When databases are linked across organizations, the issue of how to protect the privacy and confidentiality of such sensitive information is crucial to successful application of record linkage.

In this paper we present an overview of techniques that allow the linking of databases between organizations while at the same time preserving the privacy of these data. Known as 'privacy-preserving record linkage' (PPRL), various such techniques have been developed. We present a taxonomy of PPRL techniques to characterize these techniques along 15 dimensions, and conduct a survey of PPRL techniques. We then highlight shortcomings of current techniques and discuss avenues for future research.



## 1. Introduction

In recent times the world has seen an explosion in the volume of data that is being collected by organizations as well as individuals. Much of these data are about people, or they are generated by people. Examples of the former include financial data such as shopping transactions, telecommunication records, or electronic health records. Examples of the latter include emails, tweets, blog posts, and so on. It has been recognized that analyzing large data collections through the use of data mining and analytics technologies can provide a competitive edge to a commercial enterprise, can allow improved crime and fraud detection, can lead to better patient outcomes in the health sector, and can be of vital importance to national security [1].

At the same time, the quality of much of the data collected is posing a serious impediment to effective and accurate data analysis [2–4]. If the data used for data mining are of low quality, i.e., contain erroneous, missing, or out of date values, then the outcomes of a data analysis based on these data are generally also of low accuracy. As we will discuss in more detail later on, personal information, such as names and addresses, is especially prone to variations and errors [5].

One way to improve data quality and allow more sophisticated data analysis and mining is to integrate data from different sources. Integrating data allows the identification (and possible automatic correction) of conflicting data values, the enrichment of data, or the imputation of missing values [6]. The analysis of integrated data can, for example, facilitate the detection of adverse drug reactions

* Corresponding author. Tel.: +61 2 6125 7060.
*E-mail addresses:* dinusha.vatsalan@anu.edu.au (D. Vatsalan), peter.christen@anu.edu.au (P. Christen), verykios@eap.gr (V.S. Verykios).

0306-4379/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.is.2012.11.005

in particular patient groups, or enable the accurate identification of terrorism suspects [7,8].

The three major tasks in data integration are *schema matching* [9], *record linkage* [6,10], and *data fusion* [11,12]. The first task is concerned with identifying which attributes in two or more database tables contain the same information; the second task aims to identify all records that refer to the same real-world entities in two or more databases; and the third task merges pairs or groups of records that have been identified as referring to the same real-world entities into a single clean record [13]. Record linkage can also be applied on a single database to detect duplicate records [8,14]. Generally, record linkage is a challenging task because unique entity identifiers (keys) are not available in all the databases that are linked. Therefore, the common attributes available need to be used for the linkage. For databases that contain personal information about people, these common attributes generally include names, addresses, dates of birth, and other details. Three major challenges can be identified for record linkage:

- *Linkage quality*: It is commonly accepted that real-world data are 'dirty' [15], which means they contain errors, variations, values can be missing, or can be out of date. Therefore, even when records that correspond to the same real-world entity are being compared using the values of their personal identifying details, the variations and errors in these values will lead to ambiguous matches [5]. The exact comparison of personal identifying attribute values is therefore not sufficient to achieve accurate linkage results. Approximate matching as well as accurate classification techniques are needed to achieve accurate linkage quality in record linkage applications [5,16].
- *Scalability*: The number of potential comparisons required between records equals the product of the size of the two databases that are being linked. The linkage of two databases therefore has a computation complexity that is quadratic in the size of the databases. This is a major performance bottleneck in the record linkage process, because the detailed comparison of record pairs requires expensive similarity comparison functions [17,18]. The increasing size of today's databases makes the comparison of all record pairs impossible. To overcome this challenge, specific indexing techniques have been developed which remove record pairs that obviously correspond to non-matches (i.e., refer to different entities) while they maintain candidate pairs that potentially will be matching (i.e., refer to the same entity) [19].
- *Privacy and confidentiality*: When personal information about people is used in the linking of databases across organizations, then the privacy of this information needs to be carefully protected. Individual databases can contain information that is already highly sensitive, such as medical or financial details of individuals. When linked, detailed information about individuals that is even more revealing might become available, such as for people who have certain chronic diseases and who also have financial problems.
  On the other hand, when confidential business data, such as lists of suppliers or customers (which can again be businesses) or financial details, are being linked across organizations for business collaborations, then the confidentiality of such data also needs to be protected. Confidential linked data might for example reveal the amount a business owes to all its suppliers. Even the unlinked individual databases will likely contain confidential information which cannot be passed on to other organizations.
  In certain situations, it is therefore paramount that the data which are used for record linkage across organizations, as well as the results of such a linkage, are being kept secure [20].

This paper contributes a survey of historical and current techniques for privacy-preserving record linkage (PPRL). We develop a taxonomy for PPRL consisting of 15 dimensions that characterize PPRL techniques. These 15 dimensions are grouped into five topics: privacy aspects, linkage techniques, theoretical analysis, evaluation, and practical aspects. Each of these five topics consists of three dimensions. After discussing these 15 dimensions, we provide a detailed review of existing PPRL techniques, and we show how they fit into our taxonomy. Based on our review, we are able to identify gaps in existing techniques, which allows us to highlight future research directions.

The rest of this paper is structured as follows. First, in the following section, we provide an overview of application areas where record linkage has been applied, and we illustrate the importance of privacy and confidentiality within the record linkage process through a series of real-world scenarios. In Section 3 we then present the general record linkage process to provide the reader with the necessary background required to understand our taxonomy of PPRL. We define the problem of PPRL and provide an overview of the challenges involved in PPRL in Section 4. In Section 5 we describe the 15 dimensions we identified that allow us to characterize PPRL techniques. In Section 6 we then survey existing PPRL techniques, and describe how they fit into our taxonomy. We summarize the characteristics of all surveyed techniques in Table 1. We then discuss directions for future research in Section 7, and we conclude this paper in Section 8 with a summary of our findings.

## 2. Applications of record linkage

Linking records from different databases with the aim to improve data quality or enrich data for further analysis and mining is occurring in an increasing number of application areas including healthcare, government services, crime and fraud detection, and business applications.

Many health researchers are interested in aggregating health databases from different organizations for quality health data mining such as epidemiological studies or to investigate adverse drug reactions [21]. Linked health databases can also be used to develop health policies in a more efficient and effective way compared to the use of small-scale and time-consuming survey studies which traditionally have been used for this purpose [22,23].

Record linkage techniques are being used by national security agencies and crime investigators to effectively identify individuals who have committed fraud or crimes [24–26]. Many businesses take advantage of record linkage techniques for deduplicating their list of customers, which helps them to reduce the cost of running an advertising campaign or conducting other types of marketing activities. Businesses which collaborate often need to link records across their databases for successful collaborations.

Another application of record linkage is the linking of census data to provide an easy platform for compiling data for different studies, which can then be further analyzed statistically [27]. Record linkage techniques can also be applied to Web pages to identify documents that are about the same topic or are written by the same author, and to detect plagiarism in document collections [28].

When record linkage is applied within a single organization (i.e., only data owned by the same organization are linked), then generally privacy and confidentiality are not of great concern (assuming there are no internal threats). However, when data from several organizations are linked, then privacy and confidentiality need to be carefully considered, as the following scenarios illustrate.

- *Public health research*: Assume a group of public health researchers aim to investigate the types of injuries caused by car accidents, with the objective to uncover correlations between types of accidents and the resulting injuries [7]. Such research can have significant impact on policy changes that potentially save many lives [22]. This research requires data from hospitals, the police, as well as public and private health insurers. Neither of these parties is willing or allowed by law to provide their databases to the researchers. The researchers only require access to some attributes of the records that are matched across all the different databases, such as the medical details and basic biographic information, like age and gender of people who were involved in car accidents.
- *Health surveillance*: Preventing infectious diseases early before they are spread widely around a country is important for a healthy nation. Such prevention can be done by continuously monitoring early occurrences of infectious diseases. Such early outbreak detection systems require data from several sources to be collected and linked on an ongoing basis, such as human health data, consumed drugs data, and animal health data [20]. Privacy concerns arise when such data are linked and stored at a central location. Techniques are needed to ensure that private patient data as well as the confidential data collected from healthcare organizations are kept confidential and secure.
- *Serious and organized crime*: Imagine a national crime investigation unit which is tasked with fighting against crimes that are of national significance, such as organized crime syndicates. Such a unit will likely manage various national databases which draw from many different sources, including law enforcement agencies, Internet service providers, and financial institutions. Such data are highly sensitive. The collection of such data in one place for retrieval and analysis makes them vulnerable to both outsider attacks and internal adversaries, such as employees who access certain records without authorization. Generally employees are asked by the organization to sign disclosure agreements for accessing confidential data in order to reduce internal threats. Employing techniques that facilitate linking without the need of all data being given to the crime investigation unit would mean that only linked records (such as those of suspicious individuals) are available to the unit. This would significantly reduce any risks of privacy and confidentiality breaches.

## 3. The record linkage process

Record linkage is a complex process consisting of several steps [8,19], as Fig. 1 illustrates. The first step of data pre-processing (data cleaning and standardization) is crucial for quality record linkage outcomes, because most real-world data contain noisy, incomplete and inconsistent data [2,4]. This step includes filling in missing data, removing unwanted values, transforming data into well defined and consistent forms, and resolving inconsistencies in data representations and encodings [29].

The second step in record linkage is indexing [19], which is aimed at reducing the number of comparisons that need to be conducted between records by removing as many record pairs as possible that are unlikely to correspond to matches [17]. Only pairs that are potentially matching, the so-called ‘candidate record pairs’ among which we expect to find matches, are brought together to be compared in detail in the next step, the

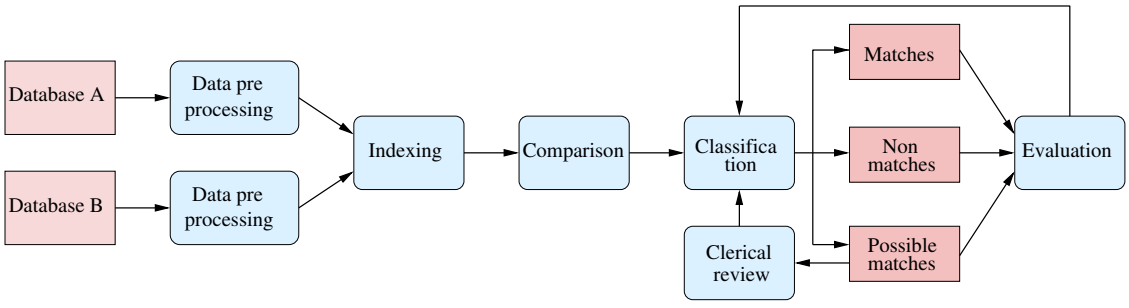



**Fig. 1.** Outline of the general record linkage process as discussed in detail in Section 3.

comparison step. The record pairs that are excluded by an indexing technique are classified as non-matches without being compared explicitly. The process of indexing is discussed further in Section 3.1.

Candidate record pairs are compared in detail in the comparison step using a variety of similarity functions [30]. If a linkage is based on using name and address details, for example, then approximate string comparison functions need to be employed which take typographical errors and variations into account [5,31]. Linkage based on date, age and numerical values needs to employ comparison functions specific to such data [32]. Section 3.2 describes several popular comparison techniques in more detail, including those that have been employed for PPRL. Several attributes are normally used when candidate record pairs are compared, resulting in a vector that contains the numerical similarity values of all compared attributes.

In the classification step, the similarity vectors of the compared candidate record pairs are given to a decision model which will classify record pairs into matches (where it is assumed that the two records in the pair correspond to the same entity), non-matches (where it is assumed that the two records in the pair correspond to different entities), and possible matches (where the classification model cannot make a clear decision) [10,18,33]. Various classification techniques have been developed for record linkage, and Section 3.3 discusses these in more detail.

If record pairs are classified as possible matches, a clerical review process is required where these pairs are manually compared and classified into matches or non-matches [34]. This is usually a time-consuming and error-prone process which depends upon experience of the experts who conduct the review. The manually classified record pairs can also be used as training data for training supervised classification techniques [19]. Alternatively, collective entity resolution techniques [35,36] can be employed that analyze not only attribute values of records but also relationships between records to determine the match status of pairs or groups of records.

Measuring the complexity, completeness, and quality, in a record linkage study is the final step in the record linkage process before the results of a linkage study can be used in an application, or the linkage approach can be implemented into an operational system. A variety of evaluation measures have been proposed to assess the complexity [19] and the quality of a linkage [18]. More details of these measures are provided in Section 3.4. In practice, measuring linkage quality is often difficult, because in many real-world record linkage applications no truth data that contain the known true match status of record pairs are available that can be used to assess linkage quality [18]. As we will discuss further in Section 7, this is especially the case for PPRL.

In the following we discuss the steps of the record linkage process in more detail, and present techniques that have been used in each of the steps. As we will discuss in Sections 5 and 7, however, many of the state-of-the-art techniques developed for record linkage in the past few years have not been investigated so far within a privacy-preserving context. Illustrating the gaps between current record linkage techniques and PPRL techniques will help to identify future research directions for PPRL.

### 3.1. Indexing

If the two database tables $A$ and $B$ which are to be linked contain $N_A$ and $N_B$ records, respectively, then potentially each record from $A$ has to be compared with all records from $B$, resulting in $N_A \times N_B$ comparisons. This becomes the major performance bottleneck in the record linkage process, since expensive detailed comparisons between records are required [17,18]. In large databases, comparing all pairs of records is therefore not feasible. It is also not necessary, because the majority of these comparisons corresponds to non-matching records [19].

To reduce this large number of potential record pair comparisons, some kind of filtering of the unlikely matches can be performed. Techniques that accomplish this are generally known as indexing, searching, or blocking techniques [17,19]. A single record attribute, or a combination of attributes, commonly called the 'blocking key', is used to decide into which blocks (or clusters) to insert a record. Records that have the same value for the blocking key will be grouped into the same block, and candidate record pairs are generated only from records within the same block. These candidate record pairs are then compared in detail in the comparison step. Applying such indexing reduces the complexity of the comparison step, since it removes many record pairs that likely correspond to non-matches without requiring their expensive detailed comparisons.

However, because real-world data contain typographical errors and other variations, there is a danger that a record is inserted into the wrong block or cluster if the attribute values used as blocking key contain an error or a variation. Therefore, (phonetic) encoding functions, such as Soundex, NYSIIS or Double-Metaphone [5], are often used to group records that have similar (sounding) values into the same block. A drawback of these phonetic encodings, however, is that they are language dependent. Limited work has been done on non-English phonetic encodings [37,38].

Indexing has a trade-off between the computational complexity and the quality of the generated candidate record pairs [17]. Having many small blocks or clusters generated based on a more specific blocking key definition will result in a smaller number of candidate record pairs and thus reduces the computation cost (though communication cost will be increased with many blocks due to the start-up costs). At the same time it is more likely that true matches are being missed. On the other hand, a less specific blocking key definition will lead to larger blocks and more candidate record pairs, but likely also to more true matches that are found [19]. In Section 3.4 we will present measures that allow this trade-off to be assessed.

Various indexing techniques for record linkage have been developed in recent years, and several surveys of these techniques have been presented [17,19,39]. In the traditional standard blocking approach used since the

1960s [10], all records that have the same blocking key value will be inserted into the same block, and only the records within the same block will be compared with each other in detail in the comparison step. Each record will be inserted into one block only.

Mapping based indexing [40] is a technique where the blocking key values are mapped to objects in a multi-dimensional Euclidean space whereby the similarities (or distances) between the blocking key values are preserved. A clustering or nearest-neighbor approach is then applied on these multi-dimensional objects to extract candidate record pairs.

One popular indexing technique is the sorted neighborhood approach [15,41], where the database tables are sorted according to a 'sorting key' over which a sliding window of fixed size is moved. Candidate record pairs are then generated from the records that are within the current window.

To overcome the issues with data that are of low quality, q-gram based indexing techniques can be used that insert each record into several blocks by generating variations of the record's blocking key value through the use of q-grams (sub-strings of length $q$ characters) [17,19]. Related to q-gram based and sorted neighborhood indexing is suffix array based indexing [42,43], where suffixes are generated from the blocking key values, and blocks are extracted from the sorted array of suffix strings.

Canopy clustering is another technique that is similar to q-gram based indexing [16,44], as an inverted index structure based on q-grams is used together with Jaccard or TF-IDF/Cosine similarity to efficiently generate overlapping clusters (called 'canopies') such that each record is inserted into several clusters. Each cluster then forms one block from which candidate record pairs are generated.

### 3.2. Comparison

Comparisons between two records can be conducted either at the record level or at the attribute (field) level. Record level comparisons concatenate the attribute values in a record into one long string, and then compare these long strings between records. With comparisons at the attribute level, comparisons are conducted between individual attribute values, with specialized comparison functions used depending upon the type of data in these attributes.

The comparison of values can either be done exact or approximate. With the former approach, a comparison function simply measures whether the values in two attributes are the same or different. Approximate comparison functions, on the other hand, measure how similar the values in two attributes are with each other. In many real-world record linkage scenarios it is not possible to simply compare two strings exactly because they can contain typographical errors and variations [5,41].

The development of approximate comparison functions, especially for string values, that can deal with variations and (typographical) errors has been a major research area in computer science [45–48]. Approximate matching of values requires a function that represents similarity as a numerical value. Generally, exact agreement is represented as a similarity of 1, total disagreement as a similarity of 0, and partial agreements as similarity values in-between 0 and 1. Many approximate comparison functions have been developed for different types of data [13]. In the following, popular techniques for approximate string comparison are described in more detail.

The Levenshtein edit-distance [47] is a commonly used comparison method for approximate string and sequence matching. It calculates the smallest number of edit operations (character inserts, deletes and substitutes) that are required to convert one string into another. Various modifications and extensions of the basic edit distance approach have been developed. Some allow for different costs of different types of edits, while others allow for gaps, or they are optimized for certain types of data. Two surveys of edit-distance based approximate string comparison functions can be found in [46,47].

Another type of comparison function is based on the idea of comparing the sub-strings, known as q-grams, that two strings have in common [49–51]. The strings to be compared are first split into shorter sub-strings of length $q$ characters using a sliding window approach, then the number of q-grams that occur in both strings is counted. Three different normalized similarity scores can then be calculated using the overlap, Dice, or Jaccard coefficient [5,13].

One approximate string comparison technique that is commonly used in record linkage applications where names and addresses need to be compared is the Jaro-Winkler approach [52,53]. This technique was developed at the US Bureau of the Census based on the expertise gained in conducting large record linkage projects. The Jaro technique combines an edit-distance and a q-gram based approach [52] by counting the number of common and transposed characters in two strings. Winkler later added several improvements to this basic comparison function [53,54], such as increased similarity if the beginning of two strings is the same, or weight adjustments based on the lengths of two strings and how many similar characters they contain.

The SoftTF-IDF string comparison technique developed by Cohen et al. [31] is aimed at the comparison of strings that contain several words. It can therefore be used for record level comparisons. The idea is based on the concepts of term frequency (TF) and inverse document frequency (IDF), as used in information retrieval, to give weights to words according to their overall occurrence in a database. Pairs of words in two strings that have a high similarity are selected to calculate an overall similarity between the two strings.

### 3.3. Classification

Assuming $k$ attributes have been compared, the outcome of the comparison step is a vector of similarity values (this is typically called 'comparison vector'), $[s_1, \ldots, s_k]$, for each candidate record pair. These vectors are used to classify record pairs as matches, non-matches, and possible matches, depending upon the decision model used [33]. Record linkage classification techniques can be broadly

grouped into four categories: threshold-based, probabilistic, rule-based, and machine learning based.

Threshold-based classification provides a simple way to classify record pairs based on the calculated overall similarity value of a pair [55]. The similarity values contained in the comparison vector are summed into a single overall similarity, $S = \sum_{i=1}^{k} s_i$, for each candidate record pair. This similarity value is then used to determine into which class a record pair belongs to based on one or two threshold values. If a single threshold $t$ is used, then all record pairs with $S \geq t$ are classified as matches, while all pairs with $S < t$ are classified as non-matches. With two thresholds, $t_l$ (lower) and $t_u$ (upper), matches are those pairs that have $S \geq t_u$, non-matches are those with $S \leq t_l$, and possible matches are those pairs that have $t_l < S < t_u$.

A widely used approach to record linkage classification is the probabilistic method developed by Fellegi and Sunter in the 1960s [10]. In this model, the likelihood that two records correspond to a match or non-match is modelled based on a priori error estimates in the data, as well as frequency distributions of individual attribute values, and the approximate similarities $s_i$ calculated in the comparison step [13]. Two thresholds (as described above) are calculated by a priori error bounds on false matches and false non-matches [10]. Extensions to the basic Fellegi and Sunter approach include the use of the Expectation Maximization (EM) algorithm to estimate the conditional probabilities required by the method in an unsupervised fashion [54,56–58].

Rule-based classification techniques are also known as deterministic techniques [59]. They use sets of rules to classify record pairs. Generating rules is often a time-consuming and complex process, since it requires manual efforts to build rule systems and also to maintain them. Logical operations (AND, OR, NOT) are used to combine several individual conditions applied on different attributes to build the complete set of rules. These rules are then applied on the comparison vectors to classify candidate record pairs into matches, non-matches, or possible matches (if desired) [14,15,60].

To accurately classify the compared candidate record pairs into matches and non-matches, many recently developed classification techniques for record linkage employ supervised machine learning approaches [8,61,62] that require training data with known class labels for matches and non-matches to train a decision model. Once trained, the model can be used to classify the remaining unlabelled pairs of records. Support vector machines and decision trees are two popular supervised learning techniques that have been employed for record linkage [55,61,62]. One limitation with supervised learning techniques is, however, that they require training data, which are not always available in record linkage applications, especially in privacy-preserving settings [13].

An alternative is to employ unsupervised learning techniques, such as clustering, which do not require training data to classify record pairs [14]. Clustering groups record pairs that are similar (according to their comparison vectors) such that each cluster consists of the records that refer to one real-world entity [16,44]. Recently developed collective [35,36], group [63], and graph-based [14,64] classification techniques, while achieving high linkage quality, are not scalable to very large databases due to their quadratic or higher computational complexity.

### 3.4. Evaluation

Evaluating the performance of record linkage algorithms in terms of how efficient and effective they are is the final step in the linkage process. The efficiency of the linkage provides a measure of how scalable a linkage technique is on large real-world applications with potentially millions of records, while the effectiveness of a linkage exercise is measured by the accuracy of the classification model used. A variety of evaluation measures has been proposed that can be used to assess the scalability [18,19] and quality [18] of the linkage process.

Scalability can be evaluated using measures that are dependent on the computing platform and networking infrastructure used, or measures that are based on the number of candidate record pairs generated. The first category of measures includes run time, memory space, and communication size, while in the second category three different measures have been proposed (as described below).

*Reduction ratio* [61] provides a value that indicates by how much an indexing technique is able to reduce the number of candidate record pairs that are being generated compared to all possible record pairs. A higher reduction ratio value means an indexing technique is more efficient in reducing the number of candidate record pairs that are being generated. If the number of true matches and true non-matches included in the candidate record pairs generated by an indexing technique are denoted with $B_M$ and $B_N$, and the total number of true matches and true non-matches in the full record pairs by $N_M$ and $N_N$, respectively, then reduction ratio is calculated as $rr = 1.0-((B_M+B_N)/(N_M+N_N))$.

*Pairs completeness* [61] measures the effectiveness of an indexing technique in the record linkage process. It is calculated as $pc = B_M/N_M$. Pairs completeness is similar to the recall measure as used in information retrieval and discussed below [65]. The third measure, *pairs quality*, measures the efficiency of an indexing technique and is similar to the precision measure discussed below. It is calculated as $pq = B_M/(B_M+B_N)$. The aim of indexing is to achieve high values for both $pc$ and $pq$, while also having a high value for reduction ratio ($rr$).

The quality of a linkage can be measured by using the metrics commonly employed in both information retrieval, and in machine learning and data mining [66,67]. These measures are defined by using four numbers as described in the following. True positives (TP) are the true matching record pairs that are correctly classified as 'matches', while false positives (FP) are the true non-matching record pairs that are classified as 'matches'. Similarly, true negatives (TN) are the true non-matching record pairs that are correctly classified as 'non-matches', and false negatives (FN) are the true matching record pairs that are classified as 'non-matches'.

Based on these four numbers, various measures can be defined. *Accuracy* is the fraction of record pairs that are correctly classified by a decision model: $acc = (TP + TN)/(TP + FP + TN + FN)$, while the *error rate* is the fraction of record pairs that are misclassified: $err = 1.0 - acc$. *Precision* is the fraction of record pairs classified as matches by a decision model that are true matches: $prec = TP/(TP + FP)$. *Recall* (also called *sensitivity* [68]) is the fraction of true matches that are correctly classified as matches by a decision model: $rec = TP/(TP + FN)$. *Specificity* is the fraction of true non-matches that are correctly classified as non-matches by a decision model: $spec = TN/(TN + FP)$. The *F-measure* or *F-score* is the harmonic mean of precision and recall, calculated as $fmeas = 2 \times ((prec \times rec)/(prec + rec))$.

Accuracy is not a suitable measure of linkage quality because classifying record pairs is generally a very imbalanced classification problem with many more non-matching record pairs compared to matching pairs [18]. The number of true non-matches can significantly distort the accuracy measure. Precision, recall and the *F*-measure are more suitable for measuring record linkage quality [13].

## 4. An overview of PPRL

As the scenarios in Section 2 have shown, the exchange of private or confidential data between organizations is often not feasible due to privacy concerns, legal restrictions, or because of commercial interests. Databases from different organizations therefore need to be linked in such ways that no sensitive information is being revealed to any of the parties involved in a cross-organizational linkage project, and no adversary is able to learn anything about these sensitive data. The increasing need of being able to link large databases across organizations while, at the same time, preserving the privacy of the entities stored in these databases, has led to the development of a new research area called *privacy-preserving record linkage* (PPRL) [69–71]. Alternative names for PPRL include *privacy record linkage* [72–75] and *blind-folded record linkage* [71,76].

The requirements of PPRL are that at the end of a linkage project only a limited amount of information is being revealed either to the parties that conducted the linkage or to another party (such as a researcher) that requires the linked data. The information revealed can either be (1) the number of records that have been classified as matches, (2) the identifiers of these matched records, or (3) a selected set of attributes from these matched records. We formally define the problem of PPRL as follows.

Assume $\mathbf{O}_1, \ldots, \mathbf{O}_m$ are the $m$ owners of their respective databases $\mathbf{D}_1, \ldots, \mathbf{D}_m$. They wish to determine which of their records $r_1^i \in \mathbf{D}_1$, $r_2^j \in \mathbf{D}_2, \ldots, r_m^k \in \mathbf{D}_m$ match according to a decision model $C(r_1^i, r_2^j, \ldots, r_m^k)$ that classifies record pairs into one of the two classes **M** of matches, and **U** of non-matches. $\mathbf{O}_1, \ldots, \mathbf{O}_m$ do not wish to reveal their actual records $r_1^i, \ldots, r_m^k$ with any other party. They however are prepared to disclose to a selected party, or to an external party, the actual values of some selected attributes of the record pairs that are in class **M** to allow further analysis.

The privacy requirement in the record linkage process adds a third challenge, privacy and confidentiality, to the two main challenges of scalability and linkage quality that were discussed in Section 1. The question now arises how to conduct the steps in the record linkage process (as was shown in Fig. 1) in a privacy-preserving setting. Privacy needs to be considered in all steps of the record linkage process, making the task of linking databases across organizations more difficult. Fig. 2 outlines the record linkage process within a privacy-preserving context.

Because data pre-processing can be conducted independently at each data source, it is not part of the techniques that are required for PPRL. However, it is crucial that all data sources conduct the same data pre-processing steps on the data they will use for linking. Some exchange of information between the data sources about what data pre-processing approaches they use, as well as which attributes they have in common that are to be used for the linkage, is therefore required.

As was discussed in Section 3.1, the indexing step is crucial to make record linkage across large databases scalable. This also applies to PPRL, but indexing for PPRL needs to be conducted in such a way that no sensitive information about individual records in the databases that are linked is revealed to any party or to an external

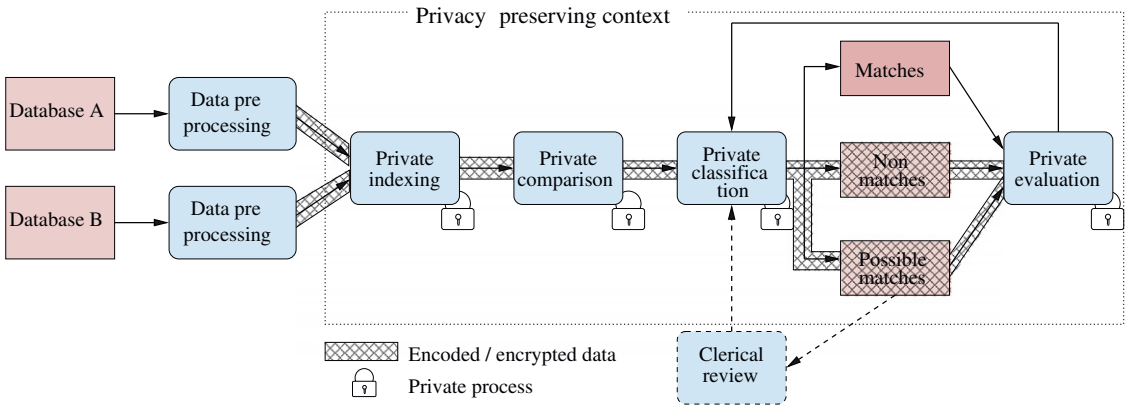



**Fig. 2.** Outline of the general privacy-preserving record linkage process as described in Section 4.

adversary. The scalability challenge of PPRL has been addressed by several recent approaches that use some form of private indexing technique, as we will discuss in Section 6.3.

The attribute values used for the comparison of records can contain variations and errors, and therefore simply encrypting these values with a standard cryptographic technique and comparing the encrypted values with an exact comparison function will not lead to high linkage quality for PPRL [71,77]. Only exactly matching attribute values can be identified with such a simple approach. A small variation in an attribute value leads to a completely different encrypted value [71]. Therefore, an approach for securely and efficiently calculating the approximate matching of attribute values is required. Several of the approximate comparison functions described in Section 3.2 have been adapted into a PPRL context.

As we discussed in Section 3.3, the output of the comparison step are the calculated similarity values for each compared record pair. These similarity values are used to classify record pairs into matches, non-matches, or possible matches. In a PPRL context, this classification needs to be conducted in such a way that no party learns anything about the records in the other parties' databases that do not match, such as similarity values for certain attributes of individual record pairs, which record pairs have low similarities, or even the distribution of similarity values across all compared record pairs. The only information to be revealed at the end of the classification step are the (number of) record pairs that have been classified as matches. How the classification techniques presented in Section 3.3 have been applied in PPRL solutions will be described in Section 6.

The evaluation of linkage quality in a privacy-preserving context is challenging, because in PPRL access to the actual record values is unlikely to be possible as this would reveal private or confidential information about these records. How to evaluate linkage quality using any of the measures presented in Section 3.4 is still an open challenge, as we will discuss further in Section 7.

### 4.1. Previous PPRL surveys

Several surveys on privacy-preserving string matching have been presented in the literature [69,78–80]. Trepetin [80] theoretically analyzed four different anonymized string matching techniques and concluded that many existing techniques fall short in providing a sound solution either because they are not scalable to large databases, or because they are unable to provide both linkage quality and privacy guarantees.

Similar conclusions were also drawn in [69,79], that survey several existing techniques for private matching ranging from classical record matching techniques enhanced by SMC techniques to provide privacy, to advanced solutions developed specific to solve the PPRL problem.

In Durham et al.'s [78] recent survey on privacy-preserving string comparators, six existing comparators that can be used in PPRL for private comparison have been experimentally evaluated in terms of their complexity, correctness, and privacy.

While all these surveys analyze and compare several private comparison functions, our survey is the first to develop a taxonomy that characterizes all aspects of PPRL, and to provide a comprehensive analysis of current approaches to PPRL.

## 5. A taxonomy of PPRL techniques

In this section we describe a taxonomy for PPRL techniques. Our aim in developing this taxonomy is to provide a clearer picture of current approaches to PPRL, and to identify gaps in these techniques which will help us to identify directions for future research. We describe 15 dimensions of PPRL which we categorize into five main

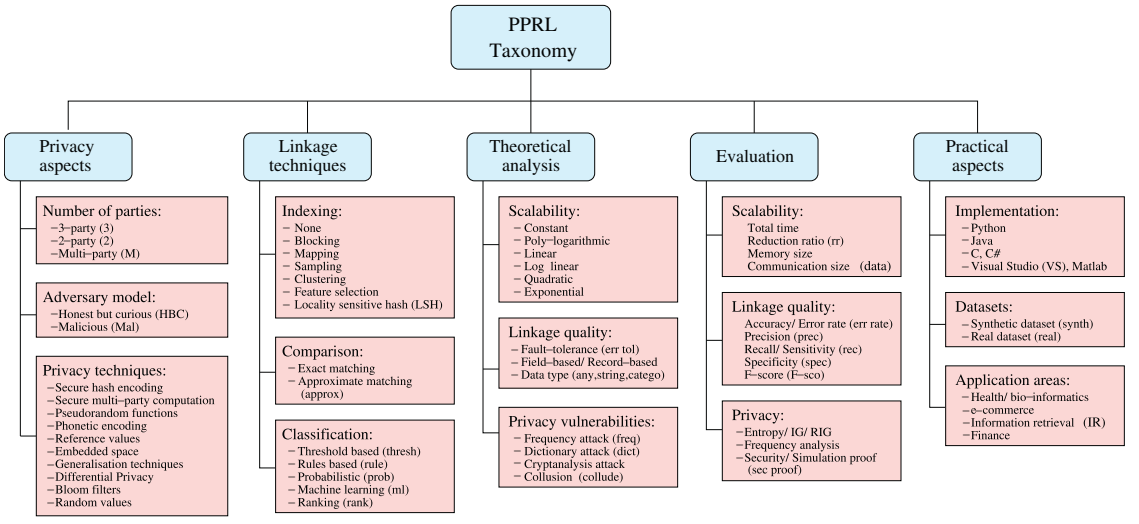



**Fig. 3.** The 15 dimensions used to characterize privacy-preserving record linkage techniques. Abbreviations shown in brackets are those used in Table 1.

topics, as is illustrated in Fig. 3. Combined, these 15 dimensions provide a comprehensive characterization of PPRL techniques. In the following sub-sections we discuss each dimension in detail, and we provide an overview of the major methodologies or techniques applied in these dimensions.

### 5.1. Privacy aspects

The privacy requirements for linking databases across organizations consider the assessment of three dimensions of PPRL techniques: how many parties are involved in a cross-organizational linkage, the adversary model assumed, and the actual techniques employed in a PPRL approach to provide privacy and confidentiality.

#### 5.1.1. Number of parties

Solutions to PPRL can be classified into those that require a third party for performing the linkage and those that do not. The former are known as 'three-party protocols' and the latter as 'two-party protocols' [7,69,81]. In three-party protocols, a (trusted) third party (which we call the 'linkage unit') is involved in conducting the linkage, while in two-party protocols only the two database owners participate in the PPRL process. The advantages of two-party over three-party protocols is that the former are more secure because there is no possibility of collusion between one of the database owners and the linkage unit, while two-party protocols often have lower communication costs. However, two-party protocols generally require more complex techniques to ensure that the two database owners cannot infer any sensitive information from each other during the linkage process.

A further characterization of PPRL techniques is if they can be extended to the efficient linking of data from more than two data sources or not. We do not consider the trivial linking of databases between all possible pairs of data sources as an efficient solution.

#### 5.1.2. Adversary model

PPRL techniques generally consider one of the two adversary models that are commonly used in the field of cryptography, and especially in the area of secure multi-party computation (SMC) [70,82,83].

(i) *Honest-but-curious behavior* (*HBC*): HBC parties are curious in that they try to find out as much as they can about the other party's inputs while following the protocol [70,83]. The protocol is secure in the HBC perspective if and only if all parties involved have no new knowledge at the end of the protocol above 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. Note that this adversary model does not prevent parties from colluding with each other with the aim to learn about another party's sensitive information [83].

(ii) *Malicious behavior*: In contrast to HBC parties, malicious parties or adversaries can behave arbitrarily. In particular, malicious parties may refuse to participate in a protocol, not follow the protocol in the specified way, choose arbitrary values for their data inputs, or abort the protocol at any time [84]. Proving privacy under this model for evaluation of a privacy technique is more difficult compared to the HBC model, because there exist additional and potentially unpredictable ways for malicious parties to deviate from the specified steps of the protocol that are undetectable by an outside observer [82,83,85].

#### 5.1.3. Privacy techniques

A variety of privacy techniques has been employed to facilitate PPRL. The major approaches are:

(i) *Secure hash encoding*: This technique has been one of the first to be used for PPRL [86–89]. One-way hash encoding functions [90] convert a string value into a hash-code (for example '*peter*' into '*51dc3dc01ea0*') such that having access to only a hash-code will make it nearly impossible with current computing technology to learn its original string value. The Message Digest (like MD5) and Secure Hash Algorithms (like SHA-1 and SHA-2) are the most widely known and used one-way hash algorithms [91].
In order to prevent dictionary attacks, where an adversary hash-encodes values from a large list of common words using existing hash encoding functions until a matching hash-code is found, a keyed hash encoding approach can be used which significantly improves the security of this privacy technique. The Hashed Message Authentication Code (HMAC) function [91] is one such approach. Without knowing the secret key, a dictionary attack will not be successful. However, frequency attacks are still possible, where the frequency distribution of a set of hash-codes is matched with the distribution of known attribute values, such as surnames [92].
A major problem when using hash encoded values for matching is, however, that only exact matches can be found [86]. Even a single character difference in a string that is encoded will lead to a completely different hash code.

(ii) *Secure multi-party computation* (*SMC*): The basic idea of SMC is that a computation is secure if at the end of the computation no party knows anything except its own input and the final results of the computed function [82,83,93]. Yao [94] first proposed the secure two-party computation problem and developed a secure solution. Goldreich et al. [95] extended this approach to several parties, and they developed a general framework for SMC. SMC employs some form of encryption schemes to allow secure computation. The two major cryptographic encryption schemes used for secure computation in the PPRL literature are commutative [96] and homomorphic [97] encryption. Various

SMC techniques have been used in PPRL for accurate computation while preserving privacy. The secure sum, secure set union, secure set intersection, and secure scalar product, are the most commonly used SMC techniques [90,93].

(iii) *Pseudo-random functions*: A pseudo-random function (PRF) is a deterministic function $f : \{0,1\}^n \rightarrow \{0,1\}^n$ which is efficient (computable in polynomial time) and takes two inputs $x$, $k \in \{0,1\}^n$. A PRF is a secure algorithm that when given an $n$-bit seed $k$, and an $n$-bit argument $x$, it returns an $n$-bit string $f_k(x)$ such that it is infeasible to distinguish $f_k(x)$ for random $k$ from a truly random function [98]. In PPRL, PRFs that have a long period and that are not predictable can be used to generate random secret values to be shared by a group of parties [77,99,100].

(iv) *Phonetic encoding*: A phonetic encoding algorithm groups values together that have a similar pronunciation, as was described in Section 3.1. The main advantage of using a phonetic encoding is that it inherently provides privacy [101], reduces the number of comparisons and thus increases scalability [5], and supports approximate matching by its tolerance against typographical variations [5,101].

(v) *Reference values*: The use of reference values, which are common to all database owners, has been applied in several PPRL approaches [75,102–104]. Such reference lists can be constructed either with random faked values, or values that for example are taken from a public telephone directory, such as all unique surnames and town names. This list of reference values can be used by the database owners to calculate the distances between their attribute values and the reference values.

(vi) *Embedded space*: This technique is based on the idea of mapping based indexing as described in Section 3.1. The attribute values are embedded (mapped) into a metric space [75,102] while the distances between these values are preserved.

(vii) *Generalization techniques*: The idea behind data generalization techniques is to overcome the problem of re-identification of individual records by generalizing the data in such a way that re-identification from the perturbed data is not possible [105–107]. $k$-Anonymity is one data generalization technique that has been used as an effective privacy technique in PPRL [108–110]. A database satisfies the $k$-anonymity criteria if every combination of quasi-identifier attribute values is shared by at least $k$ records in the database, where quasi-identifiers are attributes that can be used to identify individual entities [105].

Another recently proposed generalization technique is binning. The similarity range is binned to allow the secure exchange of similarity values resulting from the comparison step between the database owners [104]. Bins of similarity values are used instead of the actual similarity values for comparison.

(viii) *Bloom filters*: A Bloom filter is a bit-string data structure of length $l$ bits where all bits are initially set to 0. $k$ independent hash functions, $h_1, h_2, \ldots, h_k$, each with range $1, \ldots l$, are used to map each of the elements in a set $s$ into the Bloom filter by setting $k$ corresponding bit positions to 1. The Bloom filter was proposed by Bloom [111] for efficiently checking set membership [112]. In recent times, Bloom filters have been used in PPRL for private matching of records as they provide a means of privacy assurance [113–117].

(ix) *Random values*: Adding random noise in the form of extra records to the databases that are linked is a data perturbation technique [118] which can be used to overcome the problem of frequency analysis attacks within PPRL protocols. However, when adding extra records there is generally a trade-off between linkage quality, scalability and privacy [101].

(x) *Differential privacy*: Recently, differential privacy [119,120] has emerged as an alternative to generalization techniques. Instead of sharing the perturbed databases with the corresponding parties, this privacy technique allows the parties to interact with each other's databases using statistical queries. Only the perturbed results of a set of statistical queries are then disclosed to other parties.

### 5.2. Linkage techniques

The techniques used in the different steps of the PPRL process, as illustrated in Fig. 2, determine the computational requirements and the quality of the linkage results. The dimensions under this topic cover each of the required steps.

#### 5.2.1. Indexing

The techniques employed in the indexing step to facilitate record linkage solutions that scale to very large databases become more challenging if privacy concerns have to be considered. In PPRL, there is a trade-off of the indexing step not only between accuracy and efficiency, but also privacy. Several approaches have been proposed that address the scalability of PPRL solutions by adapting existing indexing techniques, such as standard blocking, mapping based blocking, clustering, sampling, and locality sensitive hash functions, into a privacy-preserving context, as discussed in Section 6.3.

#### 5.2.2. Comparison

Linkage quality is heavily influenced by how the values in records or individual attributes are compared with each other [48]. As discussed in Section 4, the naïve approach of exact matching of encrypted values does not provide a practical solution. Several of the approximate comparison functions that were presented in Section 3.2 have been investigated from a privacy preservation perspective. These techniques will be described in detail in Sections 6.2 and 6.3. The main challenge with these techniques is how the similarity between pairs of string values held at different parties can be calculated such that neither party learns about the other party's string value.

#### 5.2.3. Classification

The decision model used in PPRL to securely classify the compared record pairs needs to be effective in providing highly accurate results, such that the number of false negatives and false positives is minimized, while at the same time preserving the privacy of all records that are not part of matching pairs. As discussed in Section 3.3, a variety of classification techniques has been developed for record linkage. Details of which classification techniques have been used in PPRL will be described for individual approaches in Section 6.

### 5.3. Theoretical analysis

Theoretical estimates for the three main factors of PPRL allow the comparison of PPRL techniques, as well as an assessment of their expected scalability to large databases, quality of linkage results, and privacy guarantees.

#### 5.3.1. Scalability

This includes the computation and communication complexities that measure the overall computational efforts and cost of communication required in the PPRL process. Generally, the big $O$-notation is used to specify the computation complexity [121]. Given $n$ is the number of records in a database, the big $O$ notation of $O(\log n)$ represents logarithmic complexity, $O(n)$ linear complexity, $O(n \log n)$ log-linear complexity, $O(n^2)$ quadratic complexity, $O(n^c)$ polynomial complexity, $O(polylog\ n)$ polynomial logarithmic complexity, and $O(c^n)$ exponential complexity, where $c > 1$.

#### 5.3.2. Linkage quality

The quality of linkage is theoretically analyzed in terms of fault-tolerance of the matching technique to data errors and variations, whether the matching is based on individual fields or whole records, and the types of data the matching technique can be applied to. Fault-tolerance to data errors can be addressed by using approximate matching or pre-processing techniques such as spelling transformations.

Records can either be compared as a whole (*record based*) or by comparing the values of individual selected attributes (*field based*), as was discussed in Section 3.2. Several approximate comparison functions have been adapted into a privacy-preserving context as presented in Sections 6.2 and 6.3.

#### 5.3.3. Privacy vulnerabilities

The privacy vulnerabilities that a PPRL technique is susceptible to provide a theoretical estimate of the privacy guarantees of that technique. The main privacy vulnerabilities include *frequency attack* and *dictionary attack* (as discussed in Section 5.1.3). Bloom filter based PPRL techniques are generally also susceptible to *cryptanalysis attacks*. As Kuzu et al. [122] recently showed, depending upon the number of hash functions employed and the number of bits in a Bloom filter, using a constrained satisfaction solver allows the iterative mapping of individual encoded values back to their original values.

Another vulnerability associated with three-party and multi-party approaches is *collusion* between parties. Parties involved in a PPRL protocol may work together to find out another party's data. The vulnerabilities of individual PPRL techniques are discussed in Section 6.

### 5.4. Evaluation

The outcomes of a PPRL technique need to be evaluated in terms of the three factors: scalability, linkage quality, and privacy.

#### 5.4.1. Scalability

The measures that were discussed in Section 3.4 can be used to assess the scalability factor of PPRL similar to those assessing the scalability of non-privacy-preserving record linkage approaches.

#### 5.4.2. Linkage quality

Assuming that truth data are available (which is not the case in many PPRL applications), the linkage quality can be assessed using any of the measures that are used for record linkage in a non privacy-preserving setting that were discussed in Section 3.4.

#### 5.4.3. Privacy evaluation

Various measures have been used to assess the privacy protection that PPRL techniques provide. Here we present the most prominent measures used.

(i) *Entropy*, *Information gain* (*IG*) and *Relative information gain* (*RIG*): Entropy measures the amount of information contained in a message [101,123]. The entropy of a discrete random variable $X$ is defined as $H(X) = \sum_{x \in X} p(x) \log_2 1/p(x)$, with $x$ being an element in $X$. The conditional entropy of a discrete random variable $Y$ given the value of the variable $X$, $H(Y|X)$ can be defined as [101,123] $H(Y|X) = -\sum_{x \in X} p(x) H(Y|X = x)$.
The entropy and conditional entropy form the basis for the IG metric [123]. IG assesses the possibility of inferring the original message $Y$, given its enciphered version $X$ [101,123]. $IG(Y|X) = H(Y) - H(Y|X)$. The lower the value for IG is, the more difficult it is to infer the original value from an enciphered value. The RIG measure normalizes the scale of IG ($0.0 \leq RIG\ (Y|X) \leq 1.0$) with regard to the entropy of the original text $Y$ [101], and is defined as $RIG\ (Y|X) = IG(Y|X)/H(Y)$. Since RIG values are normalized between 0.0 and 1.0, they provide a marginal scale for comparison and evaluation.

(ii) *Security/simulation proof*: The proof of privacy of PPRL solutions can be evaluated by simulating the solutions under different adversary models [82,83,85]. A party's view in the execution of a PPRL technique requires to be simulated given only its input and output to evaluate the privacy. If under a certain adversary model (honest-but-curious or malicious, as was discussed in Section 5.1.3) a party learns nothing from the execution except its input

and output, then the technique can be proven to be secure and private.

(iii) *Frequency analysis*: A frequency attack is the most common privacy vulnerability in PPRL techniques. The probability of re-identification of values in a PPRL technique can be used as a measure to evaluate the frequency attack of that technique.

### 5.5. Practical aspects

The final three dimensions cover practical aspects of PPRL techniques including the datasets used for experimental evaluations, how a solution was implemented, and if a proposed solution was developed with a specific application area in mind.

#### 5.5.1. Implementation

This dimension specifies the implementation techniques that have been used to prototype a PPRL technique in order to conduct its experimental evaluation. Some solutions proposed in the literature provide only theoretical proofs but they have not been evaluated experimentally, or no details about their implementation have been published.

#### 5.5.2. Datasets

Experimental evaluation on one or ideally several datasets is important for the critical evaluation of PPRL techniques. Due to the difficulties of obtaining real-world data that contain personal information, synthetically generated databases are commonly used. Several tools are available to generate data [124,125]. However, to evaluate the practical aspects of PPRL techniques with regard to their expected performance in real-world applications, evaluations should ideally be done on databases that exhibit real-world properties and error characteristics.

#### 5.5.3. Application areas

This dimension describes if a PPRL technique has been developed with a certain application area in mind, or if it is specialized to link data from a certain application area. Some of the areas targeted include healthcare, census, e-commerce, information retrieval (IR), and finance applications.

## 6. A survey of privacy-preserving record linkage techniques

Research directions for PPRL were provided in [7,20] stating the needs, problems and current approaches in this area, while various techniques have been developed addressing this research problem [69,78–80]. In this section we provide a detailed review of PPRL techniques by outlining these techniques according to the 15 dimensions of our taxonomy which we presented in the previous section. We highlight terms that relate back to our taxonomy in *italic* font. Table 1 provides an overview of the surveyed publications with regard to these 15 dimensions.

We categorize PPRL techniques into three generations according to the factors that have been considered. These three generations are (1) techniques that consider exact matching of attribute values only; (2) techniques that can conduct approximate matching to improve the quality of linkage; and (3) techniques that also address scalability while conducting approximate matching. Techniques under each of these three generations are again classified into three categories, which are three-party, two-party, and multi-party techniques. For each category, we then present PPRL techniques ordered according to the year of publication. Each technique is given an identifier composed of the first three letters of the first author and the last two digits of the year of publication, which is then used in Table 1 to identify individual publications.

### 6.1. Exact matching PPRL techniques

The first generation of PPRL techniques focus only on the exact matching of records.

#### 6.1.1. Three-party techniques

*Van00*: A secure *three-party* approach in a *HBC* adversary model, proposed by Van Eycken et al. [126] in *2000*, is based on creating a single hash pseudonym for maintaining privacy. Using secure hash encoding in a three-party approach is illustrated in Fig. 4. This approach is cost effective, but it is inappropriate in real-world applications since it can only perform *exact matching* of attribute values. In this approach, both database owners will merge the values of their linkage attributes into a single string (*record based*) which is then double-hashed using a *secure hash function* and a public key encryption algorithm in order to prevent dictionary attacks. The hash strings are then used by a third party to classify the records using a *deterministic* classification technique. Experiments conducted on *health databases* showed that the accuracy of the classification increases if the concatenated string includes the full date of birth value.

*Web12*: Similar to Van Eycken et al.'s approach, a simple heuristic method for privately linking *medical data* in a *three-party* protocol was presented by Weber et al. [76] in *2012*. The authors experimentally validated the hypothesis that using a concatenated identifier made of the first two characters of the given name and surname attributes along with the date of birth attribute as the linkage attribute provides better results in terms of *sensitivity* and *specificity* compared to performing the linkage based on the identifier consisting of patients' full names and date of birth. This approach is useful when health policies preclude the full exchange of identifiers that is commonly required by other more sophisticated algorithms.

#### 6.1.2. Two-party techniques

*Fre05*: Privacy-preserving Information Retrieval (PPIR) is a research area related to PPRL, whereby PPIR employs a single query record while PPRL employs all records as match queries. Freedman et al. [100] in *2005* presented an efficient privacy-preserving keyword search algorithm for PPIR. The proposed approach is based on a *two-party* protocol considered under both the *HBC* and the *Malicious* adversary models. Their approach uses *SMC* techniques (homomorphic encryption) and *oblivious pseudo-random*

**Table 1**
Characterization of the privacy-preserving record linkage techniques surveyed in Section 6.

| PPRL technique | Privacy aspects | | | Linkage techniques | | | Theoretical analysis | | | | | | Evaluation | | | Practical aspects | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Num. of parties | Adversary model | Techniques | Indexing | Comparison | Classification | Scalability | | Linkage quality | | | Privacy | Scalability | Linkage quality | Privacy | Implementation | Datasets | Application areas |
| | | | | | | | Comp. | Comm. | Err. tol. | Matching | Data | | | | | | | |
| Van00 | 3 | HBC | Secure hashing | None | Exact | Rule | Quadratic | Linear | × | Record | Any | Freq, collude | | Accuracy | | | Real | Health |
| Web12 | 3 | HBC | Secure hashing | None | Exact | Rule | Quadratic | Linear | × | Record | Any | Freq, Collude | | rec, spec | | | Real | Health |
| Fre05 | 2 | Both | Pseudo-random, SMC | Block | Exact | Rule | Linear | Polylog | × | Field | Any | | | | Sec proof | | | IR |
| Qua98 | 3, M | HBC | Secure hashing | Block | Exact | Prob | Quadratic | Linear | √ | Field | Any | Freq, collude | | rec, spec | | | Real | Health |
| OKe04 | 3, M | HBC | Pseudo-random, SMC | None | Exact | Rule | Quadratic | Quadratic | × | Field | Any | Collude | | | Sec proof | | | Health |
| Lai06 | 2,M | HBC | Bloom filter | None | Exact | Rule | Linear | Constant | × | Record | Any | Cryptanalysis | | | | | | |
| Kan08 | 3,M | HBC | Generalization, SMC | Block | Exact | Rule | Quadratic | Linear | × | Field | Catego | Freq, collude | rr | | Sec proof | Java | Real | Health |
| Du01 | 3 | HBC | Random value, SMC | None | Approx | Rank | Linear | Linear | √ | Field | String | Collude | | | Sec proof | | | E-commerce |
| Chu04 | 3 | HBC | Secure hashing | None | Approx | Thresh | Exponential | Exponential | √ | Field | String | Freq, collude | | | | Python | Synth | Health |
| Sch09 | 3 | HBC | Bloom filter | None | Approx | Thresh | Quadratic | Linear | √ | Record | String | Cryptanalysis | | prec, rec | | Java | Both | |
| Dur10 | 3 | HBC | Bloom filter | None | Both | Rule, prob | Quadratic | Linear | √ | Field | String | Cryptanalysis | Time | prec, rec | | | Real | Health |
| Ata03 | 2 | HBC | SMC | None | Approx | Thresh | Quadratic | Quadratic | √ | Field | String | | | | Sec proof | | | Health |
| Rav04 | 2 | HBC | SMC | Sample | Approx | Thresh | Linear | Linear | √ | Field | String | | | prec | Sec proof | | Real | |
| Li11 | 2 | Both | SMC | None | Both | Thresh | Exponential | Exponential | √ | Record | String | | Time | | Sec proof | C# | Real | |
| All05 | 3 | HBC | Secure hashing, SMC | Block | Approx | Thresh | Quadratic | linear | √ | Field | String | Freq, collude | Time | prec | | Java | Real | |
| Sca07 | 3 | HBC | Embedded space, reference value | Mapping | Approx | Thresh | Quadratic | Linear | √ | Field | String | Freq, collude | Time | prec, rec | Sec proof | Java | Real | |
| Ina08 | 3 | HBC | Generalization, SMC | Block | Approx | Thresh | Quadratic | Linear | √ | Field | Catego | Freq, collude | rr | rec | | | Real | |
| Pan09 | 3 | HBC | Reference value | Cluster | Approx | Thresh | Quadratic | Linear | √ | Field | String | Freq, collude | Time | prec, rec | | | Real | Health |
| Kar11a | 3 | HBC | Phonetic | Block | Approx | Thresh | Quadratic | Linear | √ | Field | String | Freq, collude | Time | prec, rec | IG, RIG | Java | Both | |
| Kar11b | 3 | HBC | Phonetic, random value | Block | Approx | Thresh | Quadratic | Linear | √ | Field | String | Collude | Time | Acc | IG, RIG | Java | Both | |
| Haw11 | 3 | HBC | Random value | Feature selection | Approx | Thresh | Quadratic | Linear | √ | Record | String | Collude | Time | F-sco | | Matlab VS | Real | |
| Kar12 | 3 | HBC | Generalization, reference value | Cluster | Approx | Thresh | Quadratic | Linear | √ | Field | String | Freq, collude | Time | prec, rec | | | Both | |
| Dur12 | 3 | HBC | Bloom filter | LSH | Approx | Prob | Quadratic | Linear | √ | Record | String | Collude | Time | prec, rec | Freq analysis | Perl | Real | Health |
| Son00 | 2 | HBC | Pseudo-random, SMC | Block | Approx | Rule | Linear | Linear | √ | Field | String | Freq | | | Sec proof | | | IR |
| Yak09 | 2 | HBC | Embedded space, SMC | Mapping | Approx | Thresh | Quadratic | Linear | √ | Field | String | | Time | prec, rec | | | Real | |
| Ina10 | 2 | HBC | Differential privacy, SMC | Block | Approx | Thresh | Quadratic | Linear | √ | Field | Catego | | rr | rec | Sec proof | | Real | |
| Vat11 | 2 | HBC | Reference value, Generalization | Block | Approx | Thresh | Linear | Linear | √ | Field | String | Freq | Time data | prec, rec | Freq analysis | Python | Real | |
| Vat12 | 2 | HBC | Bloom filter | Block | Approx | Thresh | Quadratic | Linear | √ | Record | String | Cryptanalysis | Time, RR memory | rec | Freq analysis | Python | Real | |
| Moh11 | 2,M | Both | Generalization | Block | Approx | ml | Log-linear | Linear | √ | Field | Catego | Freq | Time | Err rate | Sec proof | | Real | Finance |

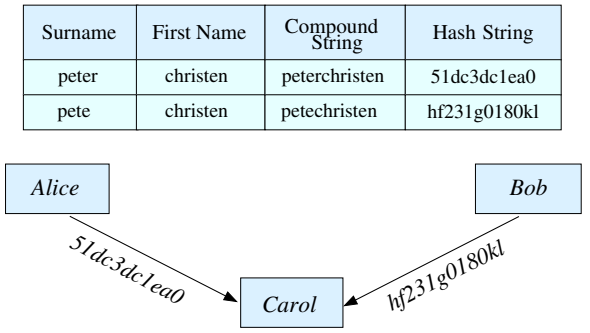



**Fig. 4.** Secure hash encoding for exact matching as used by Van Eycken et al. [126], Weber et al. [76], and Quantin et al. [86–89].

*functions*. The server holds a database of $n$ pairs $(x_i,r_i)$, each consisting of a keyword $x_i$ and its payload $r_i$. The client's input is a search keyword $w$. If there is a pair where the keyword $x_i$ is equal to the search keyword $w$ (i.e., *exact matching*), then the corresponding payload $r_i$ will be returned to the client.

In case of multiple queries where the query contains multiple search keywords, then the process is repeated for each search keyword in the query (*field based*). The server uses indexing based on *blocking* for an efficient search that defines $L$ *bins* and maps the $n$ keywords in the database into these bins using a hash function. Assume that $m$ is the maximum number of keywords mapped to any single bin, then the computation overhead of the server is $O(n)$ and the client's overhead is $O(m)$. The approach has a communication complexity which is *poly-logarithmic* in the size of the database $n$, and it needs only one round of communication.

#### 6.1.3. Multi-party techniques

*Qua98*: This is the first approach to PPRL proposed in the *1990s* by Quantin et al. [37,86–89] within the framework of *epidemiological* follow-up studies. This approach is applicable for linking more than two databases (*multi-party*) by using a *third party* for conducting the linkage. One-way *secure hash* functions such as *SHA* are used with two pads added in order to avoid dictionary attacks. The first pad is added by all the database owners to avoid dictionary attacks by the third party, and the second pad is added by the third party to prevent dictionary attacks by the database owners. Prior to hash encoding, a spelling transformation is used based on *phonetic* rules in order to reduce the impact of data errors. The comparison is limited to likely pairs of matches by using a *blocking* method based on *phonetic* encodings. Record linkage is then performed using a statistical model (*probabilistic classification*). The weight given to each linkage attribute (*field based*) is estimated by the EM algorithm. An experimental evaluation conducted using the Burgundian registry of digestive tumors and Dijon hospital discharges (both *health* data) showed a *specificity* of 100% and a *sensitivity* of 95%.

*OKe04*: In *2004*, a *multi-party SMC* based approach was proposed by O'Keefe et al. [77] for PPRL, as well as privacy-preserving extraction of a cohort of individuals' data from a database, without revealing the identity of these individuals to the database owners. This approach is applicable to perform linkage with *multiple databases* (more than two). The authors assumed an untrusted *third party*, in that the only way for the third party to obtain identifying information is through *collusion* with a database owner. The approach improves on the security and information leakage characteristics of several previous protocols, including Agrawal et al.'s [96] two-party secure intersection and equi-join protocols that use commutative encryption schemes. However, variations and (typographical) errors in the linkage attributes are not considered (the protocol only supports *exact matching*), and the protocol is more expensive than PPRL solutions that use generalization and other privacy techniques.

*Lai06*: Lai et al. [116] in *2006* proposed a *multi-party* protocol in a *HBC* adversary model that uses *Bloom filters* for private matching *without a third party* for performing the linkage. In their approach, all the records are first converted into a Bloom filter bit array (*record based*), and each party partitions its Bloom filter into the number of parties involved in the linkage and sends a segment to the corresponding party. The segments received by a party are combined using a conjunction (logical AND) operation. The resulting combined Bloom filter segments are then exchanged between the parties. Each party checks its own full Bloom filter with the results, and if the membership test is successful then it is considered to be a match. This multi-party matching is illustrated in Fig. 5.

The computation and communication costs of this approach are $O(n \times k)$ hash operations, $O(n*l^2)$ bit comparisons, and $O(l*N)$ bits of communication, respectively, where $k$ is the number of Bloom filter hash functions, $l$ is the number of bits in the Bloom filter, $N$ is the number of parties, and $n$ is the number of records in the databases. Though the cost of this approach is low since the computation is completely distributed between the parties and the creation and processing of Bloom filters are very fast, the approach can only perform *exact matching*.

*Kan08*: A *multi-party* approach based on a *generalization* technique ($k$-anonymity) for person-specific *biomedical* data was introduced by Kantarcioglu et al. [108] in *2008*. This approach performs efficient secure joins of encrypted databases by a *third party* without decrypting or inferring the contents of the joined records. It is guaranteed that each record can be linked to no less than

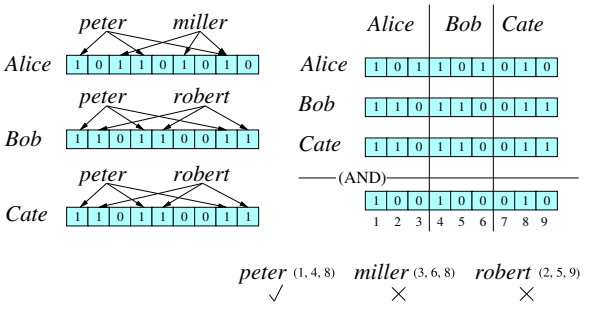



**Fig. 5.** Multi-party private record matching using Bloom filters as proposed by Lai et al. [116].

*Alice*

| Age | Postcode |
|---|---|
| 27 | 2602 |
| 60 | 3042 |
| 50 | 3021 |
| 35 | 2616 |

→

*Bob*

| Age | Postcode |
|---|---|
| [20,40] | 26** |
| [46,80] | 30** |
| [46,80] | 30** |
| [20,40] | 26** |

**Fig. 6.** k-Anonymized records ($k=2$) as used by Kantarcioglu et al. [108], Inan et al. [74], and Mohammed et al. [109].

$k$ entities in the databases. An example of this approach is shown in Fig. 6. The database owners k-anonymize their databases with the same anonymization algorithm and send the encrypted databases to the third party. When the third party performs a join, it constructs buckets corresponding to each combination of $k$-anonymous values. For each bucket, the third party performs a *secure equi-join*.

Assuming the number of buckets (*blocks*) in the databases is $g$, and the number of records in the databases is $n$, then the computation complexity of this approach is $O(n^2/g)$. Experiments conducted on the *Adult* dataset, which is publicly available on the UC Irvine Machine Learning Repository,[1] showed that the number of secure equi-joins required by the protocol is drastically reduced with a 99% *reduction ratio* when $k$-anonymous equi-join is applied, compared to the full comparison of all record pairs. The communication complexity is $O(n)$. This approach is only applicable to categorical data.

### 6.2. Approximate matching PPRL techniques

Techniques under this second generation of PPRL techniques look into the approximate matching of attribute values to remedy the problem of errors and variations in real-world data.

#### 6.2.1. Three-party techniques

*Du01*: Du et al. [127] in *2001* suggested a secure approach for private remote database access with an untrusted *third party* that is assumed to not collude with any of the two database owners. They propose four different *SMC* based *e-commerce* models for secure remote database access, all of which require privacy of customer data. The four models are the Private Information Matching (PIM), the PIM from Public Database (PIMPD), the Secure Storage Outsourcing (SSO), and the Secure Storage and Computing Outsourcing (SSCO).

*Approximate* record matching is performed using distance functions and Monte Carlo techniques. *Random values* are used to disguise the query results and intermediate results. Each value in the query string is compared individually (*field based*), and the minimum value of the final distance values of the records in the database that are compared with the query is computed to identify the closest match (*ranking*). Assuming the number of records in the databases is $n$, and the number of strings in the query is $N$, this approach only needs $O(N)$ and $O(N^2)$ communication steps for the SSO and SSCO models, respectively, while for the PIM and PIMPD models the communication cost is $O(n*N)$.

*Chu04*: A token-based *three-party* approach in a *HBC* adversary model suggested by Churches and Christen [71] in *2004* uses *hash encoded q-grams* to achieve *approximate* private linkage. Subsets of q-gram sets are used to calculate the *Dice coefficient* between attribute values (*field based* comparison). All matching hash values are compared by a third party using extra information, such as the number of q-grams contained in a subset and the total number of q-grams comprising an attribute value. A *threshold based* classification is used for deciding which record pairs are matches. This is a costly approach because of the power set generation of q-gram subsets it requires. Other drawbacks of this approach are that the comparison is done on the *full* set of record pairs, and the approach is susceptible to *frequency attacks* [80].

*Sch09*: An approach based on a combination of *Bloom filters* and *q-grams* (to facilitate *approximate matching*) was proposed by Schnell et al. [114] in *2009*. The attribute values of each record are concatenated into one string (*record based* comparison), and the q-grams of that string are mapped to one bit array (a Bloom filter) using multiple cryptographic *hash functions*. Then the Bloom filters are compared bit-wise by a *third party* in a *HBC* model, and a logical conjunction (AND) is performed on these Bloom filters to calculate the similarity according to the *Dice-coefficient*, because this similarity function is insensitive to many matching zeros in long Bloom filters (Fig. 7).

The computation cost of this approach is $O(n*q*k)$ hash operations and $O((n*l)^2)$ bit comparisons where $n$ is the size of the database, $q$ is the average number of q-grams in each record, $k$ is the number of hash functions used to map q-grams into a Bloom filter, and $l$ is the number of bits (length) in the Bloom filter. The communication cost is $O(n*l)$. This approach is efficient because of the use of Bloom filters, and it supports approximate matching of values as well, rendering it applicable to real-world conditions. However, due to the use of q-grams this approach is only applicable to matching of *string* attribute values. The approach can be attacked with a *cryptanalysis* attack as shown by Kuzu et al. [122].

*Dur10*: Durham et al. [113] in *2010* adopted Schnell et al.'s *Bloom filters* approach [114] in their work to evaluate three different PPRL approaches. They investigated *deterministic* classification techniques for *exact comparison*, *probabilistic*

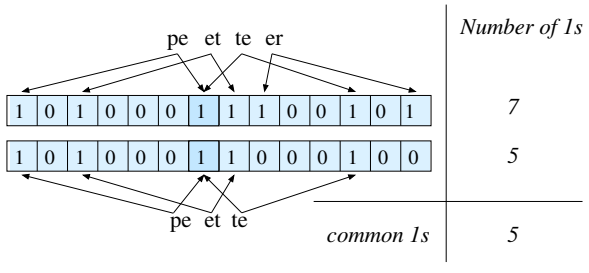



**Fig. 7.** Bloom filter mapping as used by Schnell et al. [114], Karakasidis et al. [115], and Durham et al. [113,128].

[1] www.ics.uci.edu/mlearn/

classification techniques for *exact comparison*, and *probabilistic* classification techniques for *approximate comparison*. Eleven attributes from a *clinical dataset* from the Vanderbilt University Medical Center were individually compared (*field based*) with both exact and approximate comparison (using the *Dice-coefficient*), and classified using both probabilistic and deterministic classification techniques. The empirical evaluation of these three approaches indicated that approximate comparison using probabilistic classification technique [10] outperformed the other two approaches.

#### 6.2.2. Two-party techniques

*Ata03*: A *two-party* protocol was proposed by Atallah et al. [129] in *2003* where the *edit distance* algorithm, as presented in Section 3.2, is modified for providing privacy to genome sequence *approximate* comparisons in the area of *bioinformatics*. The three types of edit operations are insertions, deletions and substitutions of characters $a$ and $b$, and each operation has an associated cost, namely $I(a)$, $D(a)$ and $S(a,b)$. The smallest overall cost of transforming one sequence into another is calculated as the edit-distance. The dynamic programming matrix $M$ is split across the two parties such that $M = M_A + M_B$ as is illustrated in Fig. 8. At each step, the minimum of three costs needs to be determined without revealing at which position the minimum occurred.

This approach allows arbitrary values for $I(a)$, $D(a)$ and $S(a,b)$. The longest common subsequence problem is a special case of the weighted edit distance problem, where insertions and deletions have unit cost, $I(a) = D(a) = 1$, and substitutions are not considered. This approach is aimed towards sequence comparisons and has a considerable communication cost. One communication step is required for each element in the matrix $M$, which is quadratic in the length of the sequences that are compared. It is therefore unsuited for tasks with large databases.

*Rav04*: In *2004*, Ravikumar et al. [130] used *SMC* techniques for secure computation of several distance functions. In their work, they presented methods for *approximate comparison* of values using string *distance metrics*, specifically TF-IDF, SoftTF-IDF and the Euclidean distance. They use a secure stochastic dot product protocol for secure computation of these distance metrics. The protocol is developed in the setting of *two parties* with a *HBC* adversary model. The use of SMC computations for achieving privacy makes the protocol computationally intensive. To overcome this drawback, they use *sampling* techniques to control the amount of communication between the two parties. Experiments on the publicly available *Cora* bibliographic dataset [13] showed high linkage quality with average *precision* of 0.85 after 1000 samples.

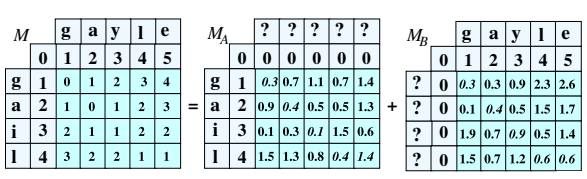



**Fig. 8.** Secure edit-distance for PPRL as proposed by Atallah et al. [129].

*Li11*: An approach for privacy-preserving group linkage (PPGL) has been introduced by Li et al. [131] in *2011* to measure the similarity of groups of records rather than individuals. A *threshold-based* PPGL method is proposed to overcome the problem of group membership inference attacks which could be employed to learn the member records of the other party's groups even though the groups are not linked. A *two-party* approach is adopted and both the *HBC* and *Malicious* adversary models are considered. Both *exact* and *approximate* comparison problems are addressed. $K$-combinations of records are first extracted from the groups (as shown in Fig. 9) and then homomorphic and commutative *SMC* encryption techniques are used to privately calculate the set intersection of the $k$-combinations.

The *Jaccard* coefficient is used at group level to calculate the similarity between two groups. In order to support approximate matching of groups of records, the *Cosine similarity* is employed in a bipartite graph to calculate the similarity of pairs of records between two groups. Both parties only learn the verdict of whether the two groups are matched or not, instead of learning the group similarity value. This approach provides privacy under the malicious adversary model as well by adopting an encrypted similarity matrix to store the intermediate results. However, the computation overhead of this approach is high. The number of comparisons required is $O(C_k^n * C_k^n)$ and the communication complexity is $O(C_k^n)$, where $C_k^n$ represents the number of $k$ combinations from $n$ records [131].

### 6.3. Scalable approximate matching PPRL techniques

In this section, we survey the third generation of PPRL techniques that address scalability to large databases while allowing the approximate matching of attribute values.

#### 6.3.1. Three-party techniques

*All05*: Al-Lawati et al. [72] proposed a secure *three-party* blocking protocol in *2005* that assumed a *HBC* adversary model for achieving high performance private record linkage by using *secure hash encoding* for computing the *TF-IDF distance measure* in a secure fashion as illustrated in Fig. 10. The approach provides *field based* and *approximate* comparison of record pairs which are then classified using a *threshold based* classification model. Token *blocking* is used for improving the computation efficiency and a third party matches the records based on the computed TF-IDF distances of the hash signatures using the *Jaccard* coefficient.

In their work, three methods have been explored to reduce the complexity of the process, which are simple

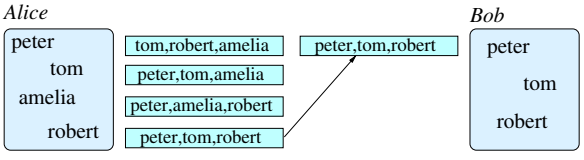



**Fig. 9.** k-Combinations extracted from Alice and Bob's groups of first name values ($k=3$) as Li et al. [131] proposed.

*Alice*

| ID | Value |
|---|---|
| a1 | {'a','b'} |
| a2 | {'c'} |

*Bob*

| ID | Value |
|---|---|
| b1 | {'b'} |
| b2 | {'a','b'} |

| | F[0] | F[1] | F[2] | F[3] |
|---|---|---|---|---|
| HS(a1) | TF IDF(a1,'b') | 0 | 0 | TF IDF(a1,'a') |
| HS(a2) | 0 | 0 | TF IDF(a1,'c') | 0 |
| HS(b1) | TF IDF(b1,'b') | 0 | 0 | 0 |
| HS(b2) | TF IDF(b2,'b') | 0 | 0 | TF IDF(b2,'a') |

**Fig. 10.** Blocking aware private record linkage using hash signatures (HS) as proposed by Al-Lawati et al. [72]. *F* is an array of floating-point numbers containing TF-IDF weights.

blocking, record-aware blocking, and frugal third party blocking [72]. Simple blocking arranges hash signatures in blocks where the similarity of a pair may be computed more than once if they are in more than one common block. Record-aware blocking solves this issue by using an identifier with every hash signature to indicate the record it belongs to. However, these methods provide a trade-off between privacy and computation and communication cost. The lowest information leakage occurs in a baseline method where no blocking is used, while the highest leakage occurs in the record-aware blocking method. The third method, the frugal third party blocking, uses a secure set intersection (SSI) *SMC* protocol to reduce the cost of transferring the whole databases to the third party by first identifying the hash signatures that occur in both databases.

*Sca07*: Scannapieco et al. [102] in *2007* presented an approach that provides privacy for both data and schema matching without revealing any information. This approach transforms records into objects in an *embedding metric space* using a set of *reference values*, while preserving the distances between record values. A *distance based approximate* comparison function is used to calculate the distances between record and reference values. These distances are then sent to a *third party*, which is assumed to follow the *HBC* behavior, to perform the linkage. To achieve secure schema matching, it is assumed that the third party holds a global schema to which the schemas of the database owners are mapped. A greedy re-sampling heuristic based on the SparseMap [132] algorithm allows the *mapping* of values into a vector space at low computational costs. However, the experimental results presented in [102] indicate that the linkage quality is affected by the greedy heuristic re-sampling method. This shows a trade-off between a more efficient building of the embedding space and the resulting quality.

*Ina08*: A hybrid approach, in a *three-party HBC* adversary model, that combines *generalization* and *cryptographic techniques* to solve the PPRL problem was proposed by Inan et al. [74] in *2008*. This method uses a *blocking* approach based on value generalization hierarchies as illustrated in Fig. 11, and the record pairs that cannot be blocked are compared in a computationally expensive *SMC* computation step using cryptographic techniques. This approach manages to perform *approximate matching* both due to the use of the *generalization* scheme in the blocking step, as well as due to the *SMC* step. The cost is reduced in the blocking step by reducing the number of candidate record pairs that need to be compared in the SMC step.

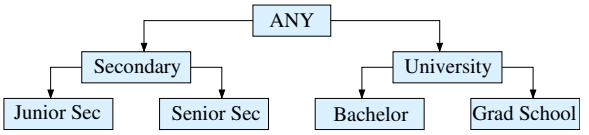



**Fig. 11.** Value generalization hierarchies as used by Inan et al. [74] and Mohammed et al. [109].

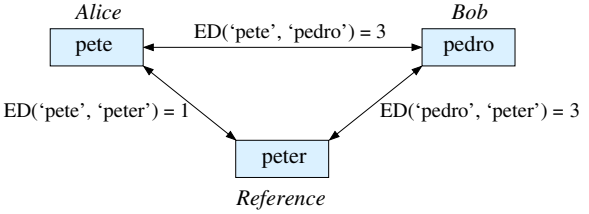



**Fig. 12.** Reference value based similarity calculation as used by Pang et al. [103]. With *ED* being the edit distance function described in Section 3.2, the triangular inequality holds: $ED(\text{'pete'},\text{'pedro'}) \leq ED(\text{'pete'},\text{'peter'}) + ED(\text{'pedro'},\text{'peter'})$.

Assuming the average number of generalized sets (blocks) generated in the blocking step is $g$, the total number of records in the database is $n$, and the number of records that remain unclassified after the blocking step is $m$ ($m < n$), then the computation and communication costs of this approach are $O(n^2/g)+O(m^2)$, and $O(g)+O(m)$, respectively. This method is however only useful with linkage attributes that can form hierarchies.

*Pan09*: Pang et al. [103] in *2009* suggested a protocol based on a set of *reference strings* that are available to both the database owners. The database owners compute the *distance* between the reference strings and their attribute values (assumed to be strings), and send the results to a *third party* that sums these distance values and finds the minimum of this sum. This process is illustrated in Fig. 12. Based on the triangular property of distance based measures [133], if this minimum distance value lies below a certain *threshold*, then the two original strings are classified as a match.

*Field based* and *approximate comparison* is performed by using a distance function such as *edit-distance* as described in Section 3.2. The parties involved in this approach are assumed to be semi-honest (*HBC*). To reduce the size of the matching space, nearest neighbor *clustering* is applied. The performance of the protocol depends crucially on the set of reference strings. The results of an empirical evaluation conducted by Bachteler et al. in 2010 [73] showed inadequate linkage quality for this approach in terms of *precision* and *recall*. Increasing the size of the reference table improves the linkage quality to some extent, but this is impractical because it leads to longer run times.

**Kar11a**: An approach to PPRL consisting of a secure *blocking* component based on *phonetic encoding* algorithms and a secure matching component where *approximate*

*matching* is performed based on *distance based* methods is presented by Karakasidis et al. [115] in *2011*. A *three-party* setting in a *HBC* model is assumed. This approach uses a secure version of the *edit distance* comparison function on a *Bloom filter* data structure. *Field based* comparisons between records are conducted and record pairs are classified using a *threshold based* model. The experimental study conducted on a *synthetic dataset* generated using the *Febrl* [30] tool showed that the approach outperforms the original edit distance algorithm in terms of complexity (due to the secure blocking component) while preserving privacy, and it also offers almost the same matching performance. The secure blocking component offers a trade-off between matching accuracy and scalability.

*Kar11b*: Karakasidis et al. [101] in *2011* proposed three different methods for phonetic based PPRL using faked *random value* injection techniques. These techniques are the Uniform Cipher Text/Uniform Plain Text, Uniform Cipher Texts by Swapping Plain Texts, and *k-anonymous* Cipher Texts. This work uses a *Soundex* [5] based fake injection strategy for private blocking, and a modified version of the Levenshtein *edit distance* for performing *approximate matching*. In the first method, fake values are added to the datasets such that both the actual values and the Soundex values exhibit uniform distributions. This increases the complexity due to massively oversized datasets. The second method overcomes this drawback by modifying the frequency of attribute values such that all Soundex values occur equally frequent. This does not create an excessive number of extra faked records as with the first method. However, the attribute values that were removed where the corresponding Soundex value had more than the average number of attribute values for each Soundex code will not participate in the linkage process.

The third method aims at creating datasets where each Soundex code reflects at least $k$ attribute values as shown in Fig. 13. The parameter $k$ is tunable to adjust the number of faked records created. This work is experimentally evaluated using a real-world Australian telephone database. It is stated that in terms of *information gain*, using a Soundex based fake injection strategy offers adequate privacy for privacy-preserving blocking [101].

*Haw11*: Hawashin et al. [134] in *2011* proposed an efficient *three-party* approach for semantic similarity joins using *long string attributes* (corresponding to *record based* comparison), such as paper abstracts, movie summaries, product descriptions, and user feedbacks. Similarities are calculated *approximately* and classification is based on a similarity *threshold*. The two database owners generate their term by long string value matrices, such that each row represents a term (word) and each column represents a long string value, and calculate TF-IDF weights. They then perform *unsupervised feature selection* using the mean of their TF-IDF values. The list of selected features along with some *random features* are sent to a third party that returns the intersection of these two feature lists.

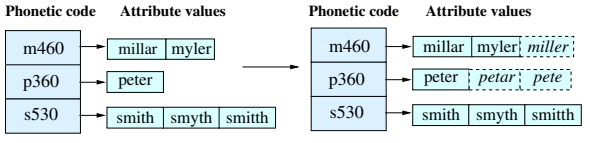



**Fig. 13.** k-Anonymous phonetic encoding ($k=3$) as proposed by Karakasidis et al. [101].

The database owners then send the selected feature values of the records with *randomly generated records* to the third party that performs the semantic join operation and classifies the pairs as matches that have a *Cosine similarity* greater than or equal to a minimum *threshold* value. Among the three semantic methods for joining, which are diffusion maps [135], latent semantic indexing [136], and locality preserving projection [137], the results of the experimental evaluation showed that the diffusion maps method provided the best performance results in terms of *F-measure* [134].

*Kar12*: In *2012* a *k*-anonymous (*generalization*) private blocking approach based on a *reference table* was proposed by Karakasidis et al. [110] for *three-party* PPRL techniques. Initially *clusters* are created for the set of reference values that are shared by the database owners and then each database owner assigns the set of blocking key values in their own data to the respective clusters. A *nearest neighbor clustering* is used for cluster creation and the *Dice-coefficient* is employed to assess the similarity between values in the sets. These clusters are then sent to a third party that merges the corresponding clusters to generate candidate record pairs. Clusters are formed from the reference table such that each cluster consists of at least $k$ elements in the reference set. Experiments conducted using a *real-world Australian telephone book* as reference table and *synthetic* data generated using the *Febrl* tool [30] as datasets validate that this approach provides *k-anonymity* privacy guarantees and supports *approximate matching* while providing blocking to reduce the number of candidate record pairs.

*Dur12*: Recently, Durham [128] proposed a framework for PPRL using *Bloom filters*. In this work the author suggested *record level* Bloom filter encoding to overcome the problem of cryptanalysis attack associated with field level Bloom filter encoding, and also used *locality-sensitive hash* (LSH) *functions* for private blocking to reduce the computational complexity. A single Bloom filter is used to encode the entire record by using weighted random bit selection from each field based Bloom filter. A *probabilistic* method based on agreement and disagreement weights is used for classification. Empirical studies conducted on *real datasets* showed that this approach outperforms existing Bloom filter based approaches.

#### 6.3.2. Two-party techniques

*Son00*: The approach of Song et al. [99] in *2000* in a *two-party* context with the *HBC* model takes into consideration the problem of *approximate matching* by calculating *enciphered permutations* of values using *pseudo-random functions* for private approximate searching of documents by certain query values. The query values can either be a single word or an advanced query with multiple words. The approximate comparison on the advanced query is processed based on individual words

(i.e., *field based*). If an encrypted value matches at least one of the enciphered permutations (*rule based*), then the pair of values can be considered as a match since the permutation occurred due to a typographical error.

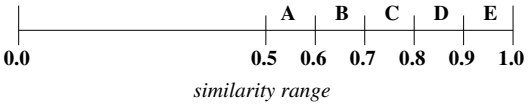
The use of an *encrypted index* data structure based *blocking* provides an efficient search when the data size is large. For a document of length $n$, it only requires $O(n)$ cipher operations and introduces almost no communication overhead. However, it is practically impossible to predict all possible permutations by pre-computing all types of errors and variations that might occur in real-world applications. The approach is also susceptible to *frequency attacks* if a certain number of words are being queried.

*Yak09*: Based on the work by Scannapieco et al. [102], a similar approach was proposed by Yakout et al. [75] in *2009* which uses Scannapieco's vector representation of attribute values and eliminates the need of a third party (*two parties* only) for performing PPRL. Complex numbers are calculated to create a *complex plane* and in the first step the likely matched pairs are computed by moving an adjustable width slab within this complex plane. *Euclidean distance* is used to measure the *approximate similarity* between records. Based on these distances, similar record pairs are classified as those that are within the slab width (*threshold based* classification). These similar pairs are compared in detail in the second step using a *SMC* based *secure scalar product* protocol based on randomized vectors.

The overall computation complexity of the first step is $O(m+2n \log n)$, where $m$ is the number of likely matched pairs and $n$ is the number of records in the databases. The second step has a computation complexity of $O(m^2)$ SMC operations. The communication complexity of this approach equals $O(n)+O(m)$. This is an improvement over Scannapieco's work in the privacy and scalability aspects.

*Ina10*: Inan et al. [119] in *2010* presented an approach for PPRL based on *differential privacy* as was described in Section 5.1.3. The approach combines *differential privacy* and *cryptographic methods* to solve the PPRL problem in a *two-party* protocol following the *HBC* adversary model. It uses specialized multi-dimensional tree index data structure based *blocking* (kd-tree, BSP-tree, $R^*$-tree, etc.) to improve scalability. Previous work presented by Inan et al. [74] focused on generalization based on $k$-anonymity to provide a scalable solution, which does not provide sufficient privacy. The work based on differential privacy provides strong privacy guarantees and a trade-off between accuracy, privacy and scalability [119]. The computation and communication costs are $O(n^2/p)+O(m^2)$ and $O(p)+O(m)$, respectively, where $n$ is the number of records in the databases, $p$ is the number of partitions generated in the blocking step, and $m$ is the number of remaining records to be compared in the SMC step.

*Vat11*: An efficient *two-party* approach for PPRL in a *HBC* adversary model for *approximate matching* was proposed by Vatsalan et al. [104] in *2011*. Similar to the approach by Pang et al. [103], their technique also uses *reference values* for securely calculating the similarities between attribute values and reference values. By using the reverse triangular inequality property of *distance metrics*, these values are compared without the need of sending them to a third party. The calculated similarity values are *generalized* into bins (see Fig. 14) to allow their secure exchange between the two database owners.




| Attribute value | Reference value | Similarity value | Bin label |
|---|---|---|---|
| amelie | amilia | 0.7 | C |
| smyth | smith | 0.9 | E |
| peter | peter | 1.0 | E |

**Fig. 14.** Binning similarity values to allow secure exchange in a two-party context as proposed by Vatsalan et al. [104].

Record pairs that have the same similarity bin combinations in their linkage attributes (*field based* comparison) or that have a similarity binning distance less than a maximum binning distance, which is calculated according to the minimum similarity *threshold* agreed on by the database owners, are classified as matches. The number of bins is a parameter that has to be chosen carefully, since it has a trade-off between scalability, privacy, and accuracy. *Phonetic encoding* is applied to *block* the databases in order to make the solution scalable to large databases. Experiments on an Australian telephone directory database with nearly two million records showed that the approach has a *linear* computation and communication scalability to large databases and achieves high accuracy by tuning the number of bins appropriately.

*Vat12*: A *two-party* approach based on the use of *Bloom filters* for *approximate* private matching was developed by Vatsalan et al. [117] in *2012*. They propose an iterative classification approach where the database owners iteratively reveal bits from their Bloom filters without compromising privacy and complexity. At each iteration they calculate the minimum similarity based on the revealed bit positions using the *Dice-coefficient*, and classify the pairs into matches, non-matches, and possible matches. The pairs that are classified as possible matches are taken to the next iteration where more bit positions are revealed to classify the pairs. A *length filtering* method is used to reduce the number of record pair comparisons. Experiments conducted on a real-world database showed that the approach is scalable and provides sufficient privacy characteristics.

### 6.3.3. Multi-party techniques

*Moh11*: Mohammed et al. [109] in *2011* proposed an approach for efficient PPRL using the $k$-anonymity based *generalization* privacy technique without the need of a trusted third party (*two parties* only). This work is based on the secure DkA framework proposed by Jiang and Clifton [138] for integrating two private data tables into a $k$-anonymous table. However, the DkA framework is not scalable to large databases. Mohammed et al. presented two scalable methods to securely integrate private data from two or more database owners (i.e., *multiple data*

*sources*) based on the two different adversary models. A top-down specialization approach is used to *generalize* the databases, as was illustrated in Fig. 11.

The database owners find the global winner candidate with the best score that provides less information to the other party according to some criteria, and then perform the specialization on that candidate for generalizing the databases. The well-known C4.5 classifier is used to recursively *block* (generalized buckets) and classify the records in the databases. The computation and communication costs of this approach are $O(n \log n)$ and $O(n)$, respectively, where $n$ is the number of records in the databases. To prevent malicious parties from sending false scores, game-theoretic concepts are used. In game theory, a rational participation is assumed where all the parties contribute equally in the generalization process, but if a party deviates from the protocol the value of the contribution will be decreased for that party. Empirical studies conducted by the authors using the real-world Adult dataset demonstrated the scalability of the solutions.

## 7. Discussion and research directions

In this section, we analyze the surveyed PPRL techniques as characterized in Table 1 with regard to the taxonomy proposed. This analysis highlights several areas of where future research in PPRL needs to focus on.

As our survey has shown, since the beginning of the development of techniques that aim to provide solutions for PPRL, there has been a large variety of techniques that have been investigated. There is a clear path of progress, starting from early techniques that solve the problem of privacy-preserving exact matching, moving on to techniques that allow approximate matching while keeping the attribute values that are matched secure, and finally in the last few years focusing on techniques that address the issue of scalability of PPRL on large databases.

### 7.1. Privacy aspects

With regard to privacy, several topics require further attention in order to make PPRL more applicable for practical applications.

- *PPRL on multiple databases*: Most work in PPRL (and record linkage in general) thus far has concentrated on linking data from two database owners only. Only a small number of approaches have investigated how to efficiently link databases from more than two organizations [77,87,108,109,116]. As the scenarios in Section 2 have shown, however, linking data from more than two sources is commonly required. Recent work by Sadinle et al. [139] extends the Fellegi and Sunter model to link more than two databases in a non-privacy-preserving context.
  In a three-party scenario, extending PPRL protocols can be accomplished such that all database owners send their data to the linkage unit, which then conducts the linkage. The linkage unit will however become the computational bottleneck as the number of parties increases and it therefore has to link more and more records.
  For two-party protocols, new approaches need to be developed when three or more parties wish to identify which records they have in common. For efficiency reasons, communication schemes such as rings or binary trees can be considered, where pairs of database owners link their databases and pass on the record pairs classified as matches to the next party.
  The possibility of collusion by a subset of parties involved in a multi-party PPRL protocol with the aim to find out about another party's sensitive data will need to be carefully considered.
- *PPRL for malicious adversaries*: Most solutions proposed so far assume the HBC adversary model. The solutions that can be used with malicious adversaries are generally more complex and mainly use SMC based techniques which have high computation and communication complexities, and are thus not scalable to large databases. The approach by Mohammed et al. [109] uses game theory concepts to deal with malicious parties, and as such provides a novel approach to PPRL. Another problem with solutions that assume the malicious model is that evaluating the privacy under this model is very difficult.
- *Privacy techniques*: As the characterization of PPRL techniques in Table 1 has shown, many different privacy techniques have been explored over the past nearly two decades to address the various challenges posed by the requirements of PPRL. More advanced privacy techniques have been developed in the second and third generations while first generation techniques are mainly based on secure hash encoding only. Some of the more commonly used techniques include Bloom filters and generalization techniques such as $k$-anonymity, however they both have their limitations. Bloom filters can only be used in the calculation of certain set-based similarity measures, while generalization techniques require some approach to conceptually generalize attribute values, which is data and domain dependent.
  The more recently developed differential privacy technique [120] is capable to provide sufficient privacy guarantees compared to other privacy techniques (except SMC techniques). More research is needed to investigate the use of differential privacy and other advanced scalable techniques that provide sufficient privacy protection to work in combination with or even replace expensive SMC based techniques.

### 7.2. Linkage techniques

Research in non-PPRL in recent years has developed various advanced techniques that provide improved scalability and linkage quality. Thus far, however, most of these techniques have not been investigated in a privacy-preserving setting.

- *Indexing*: Most work in PPRL that has investigated scalability, through some form of indexing technique, has

employed the basic standard blocking approach. As explained in Section 3.1, this technique is not efficient and has quadratic complexity when the databases are large. Mapping based indexing [40] is a second technique that has been employed in PPRL [75,102]. The use of locality sensitive hashing (LSH) has recently been proposed to improve the scalability of PPRL techniques [128]. Other efficient techniques such as the sorted neighborhood approach, or suffix-array based indexing techniques, need to be explored in a privacy-preserving setting.

- *Comparison*: Most PPRL solutions in the second and third generations consider approximate comparison. However, they are mostly applicable only to string data type. Research is required to develop approximate comparison functions that are tailored to numeric, date, age, and time attributes, and even for those containing geographic and other complex types of information [13].
- *Improved classification*: As Table 1 shows, most current approaches to PPRL employ a simple threshold or rule-based deterministic approach to classify the compared record pairs. Only limited work has been conducted that investigates the application of advanced classification techniques that have been developed for record linkage in the past decade, such as machine learning or graph-based collective classification approaches, in a privacy-preserving context [109,140,141]. This constitutes a significant gap between the state-of-the-art techniques in non-PPRL techniques and those employed in PPRL, and provides ample opportunities for research to significantly improve PPRL techniques.

### 7.3. Theoretical analysis

While the analysis of the scalability of PPRL algorithms with regard to their communication and computation requirements are based on standard approaches such as the big *O*-notation [121], and the analysis of linkage quality can be assessed by the type of data that can be matched, and if matching is exactly or approximately, the theoretical assessment of the privacy achieved within PPRL is currently the least matured theoretical aspect.

A standard set of privacy measures is required that allows the comparative theoretical analysis of privacy preservation that can be achieved by PPRL techniques. As there are often different privacy requirements in different practical applications of record linkage, a measure such as the privacy spectrum proposed by Reiter and Rubin [142] might be suitable. The privacy spectrum measures the degree of privacy attained against an adversary on a scale from 0 (absolute privacy) to 1 (provable exposure).

### 7.4. Evaluation

The evaluation of the implementation of PPRL techniques with regard to their scalability, linkage quality, and privacy preservation poses some unique challenges.

- *Assessing linkage quality and completeness*: Current PPRL techniques only address how to assess linkage quality and completeness to a very limited degree. Given in a practical linkage situation the true match status of the compared record pairs is unlikely to be known, and in a PPRL scenario even the actual record attribute values cannot be inspected (because this would reveal private information), measuring linkage quality and completeness is difficult [18,143].
  Without being able to assess linkage quality and completeness, PPRL will not be useful for real-world linkage applications, because not knowing how good the results of a linkage project are is not an option in practical applications, where linkage quality and completeness are two crucial factors for successful PPRL.
- *A framework for PPRL*: There is currently no framework available for PPRL that facilitates the comparative evaluation of different PPRL techniques with regard to privacy, scalability, and linkage quality. Researchers have used a variety of evaluation measures and datasets (both real and synthetic), which makes comparing existing techniques difficult.
  A framework for PPRL will need to facilitate the detailed specifications of all building blocks of the PPRL process in the form of abstract representations, such as XML schemas. This will make it possible for researchers to implement their novel algorithms and techniques, and integrate them so as to evaluate them comparatively. Such a framework will lead to a much improved understanding of the overall PPRL process.

### 7.5. Practical aspects

So far it seems that no single PPRL technique has outperformed all other techniques in the three aspects of linkage quality, privacy preservation, and scalability to large datasets. However, the lack of comprehensive studies that compare many existing techniques within the same framework and on many different types of data, means that it is currently not possible to determine which technique(s) perform better than others on data with different characteristics and of different sizes. Conducting such large experimental studies is one avenue of research that would be highly beneficial to better understand the characteristics of PPRL techniques.

## 8. Conclusion

In this paper we have presented a survey of historical and current state-of-the-art techniques for PPRL. We have identified 15 dimensions that allowed us to characterize PPRL techniques, and to generate a taxonomy of such techniques. This proposed taxonomy can be used as a comparison and analysis tool for PPRL techniques. Through this taxonomy we identified various shortcomings of current approaches to PPRL that suggest several future research directions in this field.

Crucially, there is currently no overarching framework available that allows different approaches to PPRL to be evaluated comparatively. Researchers have used various datasets as well as a variety of measures to evaluate the

privacy, scalability, and linkage quality, of their PPRL techniques.

Most research in PPRL so far has concentrated on the development of privacy-preserving approximate matching of strings, while only limited work has been conducted on the other steps of the record linkage process, namely indexing to make PPRL scalable to large databases, classification techniques to achieve high linkage quality, and techniques that allow both linkage quality and completeness to be evaluated. Solving these open research questions is a core requirement to make PPRL applicable for practical applications.

## References


[1] J. Han, M. Kamber, J. Pei, Data Mining: Concepts and Techniques, 3rd edition, Morgan Kaufmann, 2011.
[2] C. Batini, M. Scannapieco, Data Quality: Concepts, Methodologies and Techniques, Data-Centric Systems and Applications, Springer, 2006.
[3] Y. Lee, L. Pipino, J. Funk, R. Wang, Journey to Data Quality, MIT Press, 2009.
[4] E. Rahm, H.H. Do, Data cleaning: problems and current approaches, IEEE Data Engineering Bulletin 23 (4) (2000) 3–13.
[5] P. Christen, A comparison of personal name matching: techniques and practical issues, in: IEEE ICDM Workshop on Mining Complex Data, Hong Kong, 2006.
[6] T. Herzog, F. Scheuren, W. Winkler, Data Quality and Record Linkage Techniques, Springer Verlag, 2007.
[7] P. Christen, Privacy-preserving data linkage and geocoding: current approaches and research directions, in: IEEE ICDM Workshop on Privacy Aspects of Data Mining, Hong Kong, 2006.
[8] A. Elmagarmid, P. Ipeirotis, V.S. Verykios, Duplicate record detection: a survey, IEEE Transactions on Knowledge and Data Engineering 19 (1) (2007) 1–16.
[9] Z. Bellahsene, A. Bonifati, E. Rahm, Schema Matching and Mapping, Data-Centric Systems and Applications, Springer, 2011.
[10] I.P. Fellegi, A.B. Sunter, A theory for record linkage, Journal of the American Statistical Society 64 (328) (1969) 1183–1210.
[11] X. Dong, F. Naumann, Data fusion: resolving data conflicts for integration, Proceedings of the VLDB Endowment 2 (2) (2009) 1654–1655.
[12] J. Bleiholder, F. Naumann, Data fusion, ACM Computing Surveys 41 (1) (2008) 1–41.
[13] P. Christen, Data Matching, Data-Centric Systems and Applications, Springer, 2012.
[14] F. Naumann, M. Herschel, An introduction to duplicate detection, Synthesis Lectures on Data Management, vol. 2 (1), 2010, pp. 1–87.
[15] M.A. Hernandez, S.J. Stolfo, Real-world data is dirty: data cleansing and the merge/purge problem, Data Mining and Knowledge Discovery 2 (1) (1998) 9–37.
[16] W.W. Cohen, J. Richman, Learning to match and cluster large high-dimensional data sets for data integration, in: ACM SIGKDD, Edmonton, 2002.
[17] R. Baxter, P. Christen, T. Churches, A comparison of fast blocking methods for record linkage, in: ACM SIGKDD Workshop on Data Cleaning, Record Linkage and Object Consolidation, Washington, DC, 2003.
[18] P. Christen, K. Goiser, Quality and complexity measures for data linkage and deduplication, in: F. Guillet, H. Hamilton (Eds.), Quality Measures in Data Mining, Studies in Computational Intelligence, vol. 43, Springer, 2007, pp. 127–151.
[19] P. Christen, A survey of indexing techniques for scalable record linkage and deduplication, IEEE Transactions on Knowledge and Data Engineering 24 (9) (2012) 1537–1555.
[20] C. Clifton, M. Kantarcioglu, A. Doan, G. Schadow, J. Vaidya, A. Elmagarmid, D. Suciu, Privacy-preserving data integration and sharing, in: ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, Paris, 2004, pp. 19–26.
[21] E. Brook, D. Rosman, C. Holman, Public good through data linkage: measuring research outputs from the Western Australian data linkage system, Australian and New Zealand Journal of Public Health 32 (1) (2008) 19–23.
[22] D.E. Clark, Practical introduction to record linkage for injury research, Injury Prevention 10 (2004) 186–191.
[23] C.W. Kelman, J. Bass, D. Holman, Research use of linked health data—a best practice protocol, Australian and New Zealand Journal of Public Health 26 (2002) 251–255.
[24] J. Jonas, J. Harper, Effective Counterterrorism and the Limited Role of Predictive Data Mining, Cato Institute, 2006.
[25] C. Phua, K. Smith-Miles, V. Lee, R. Gayler, Resilient identity crime detection, IEEE Transactions on Knowledge and Data Engineering 24 (3) (2012) 533–546.
[26] G. Wang, H. Chen, H. Atabakhsh, Automatically detecting deceptive criminal identities, Communications of the ACM 47 (3) (2004) 70–76.
[27] W.E. Winkler, Overview of Record Linkage and Current Research Directions, Technical Report RR2006/02, US Bureau of the Census, 2006.
[28] M. Murugesan, W. Jiang, C. Clifton, L. Si, J. Vaidya, Efficient privacy preserving similar document detection, International Journal on Very Large Databases 19 (4) (2010) 457–475.
[29] T. Churches, P. Christen, K. Lim, J.X. Zhu, Preparation of name and address data for record linkage using hidden Markov models, BioMed Central Medical Informatics and Decision Making 2 (9) (2002).
[30] P. Christen, Development and user experiences of an open source data cleaning, deduplication and record linkage system, SIGKDD Explorations 11 (1) (2009) 39–48.
[31] W.W. Cohen, P. Ravikumar, S. Fienberg, A comparison of string distance metrics for name-matching tasks, in: IJCAI Workshop on Information Integration on the Web, Acapulco, 2003, pp. 73–78.
[32] P. Christen, Febrl: an open source data cleaning, deduplication and record linkage system with a graphical user interface, in: ACM SIGKDD, Las Vegas, 2008, pp. 1065–1068.
[33] L. Gu, R. Baxter, Decision models for record linkage, in: Selected Papers from AusDM, Lecture Notes in Computer Science, vol. 3755, Springer, 2006, pp. 146–160.
[34] W.E. Winkler, Methods for evaluating and creating data quality, Information Systems 29 (7) (2004) 531–550.
[35] I. Bhattacharya, L. Getoor, Collective entity resolution in relational data, ACM Transactions on Knowledge Discovery from Data 1 (1) (2007).
[36] D. Kalashnikov, S. Mehrotra, Domain-independent data cleaning via analysis of entity-relationship graph, ACM Transactions on Database Systems 31 (2) (2006) 716–767.
[37] C. Quantin, H. Bouzelat, L. Dusserre, Irreversible encryption method by generation of polynomials, Medical Informatics and the Internet in Medicine 21 (2) (1996) 113–121.
[38] R. Schnell, T. Bachteler, S. Bender, A toolbox for record linkage, Austrian Journal of Statistics 33 (1–2) (2004) 125–133.
[39] J. Nin, V. Muntes-Mulero, N. Martinez-Bazan, J.-L. Larriba-Pey, On the use of semantic blocking techniques for data cleansing and integration, in: IDEAS, Banff, Canada, 2007, pp. 190–198.
[40] L. Jin, C. Li, S. Mehrotra, Efficient record linkage in large data sets, in: DASFAA, Tokyo, 2003, pp. 137–146.
[41] M.A. Hernandez, S.J. Stolfo, The merge/purge problem for large databases, in: ACM SIGMOD, San Jose, 1995, pp. 127–138.
[42] A. Aizawa, K. Oyama, A fast linkage detection scheme for multi-source information integration, in: WIRI, Tokyo, 2005, pp. 30–39.
[43] T. de Vries, H. Ke, S. Chawla, P. Christen, Robust record linkage blocking using suffix arrays and Bloom filters, ACM Transactions on Knowledge Discovery from Data 5 (2) (2011).
[44] A. McCallum, K. Nigam, L.H. Ungar, Efficient clustering of high-dimensional data sets with application to reference matching, in: ACM SIGKDD, Boston, 2000, pp. 169–178.
[45] P. Hall, G. Dowling, Approximate string matching, ACM Computing Surveys 12 (4) (1980) 381–402.
[46] P. Jokinen, J. Tarhio, E. Ukkonen, A comparison of approximate string matching algorithms, Software-Practice and Experience 26 (12) (1996) 1439–1458.
[47] G. Navarro, A guided tour to approximate string matching, ACM Computing Surveys 33 (1) (2001) 31–88.
[48] E.H. Porter, W.E. Winkler, Approximate String Comparison and its Effect on an Advanced Record Linkage System, Technical Report RR97/02, US Bureau of the Census, 1997.
[49] H. Keskustalo, A. Pirkola, K. Visala, E. Leppänen, K. Järvelin, Non-adjacent diagrams improve matching of cross-lingual spelling variants, in: String Processing and Information Retrieval, Springer, 2003, pp. 252–265.
[50] K. Kukich, Techniques for automatically correcting words in text, ACM Computing Surveys 24 (4) (1992) 377–439.

[51] B. van Berkel, K. De Smedt, Triphone analysis: a combined method for the correction of orthographical and typographical errors, in: ANLP, Austin, TX, USA, 1988, pp. 77–83.
[52] M. Jaro, Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida, Journal of American Statistical Association (1989) 414–420.
[53] W. Winkler, String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage, in: Survey Research Methods, American Statistical Association, 1990, pp. 778–783.
[54] W. Winkler, Improved decision rules in the Fellegi–Sunter model of record linkage, in: Survey Research Methods, American Statistical Association, vol. 274, 1993, p. 279.
[55] P. Christen, Automatic record linkage using seeded nearest neighbour and support vector machine classification, in: ACM SIGKDD, Las Vegas, 2008.
[56] X. Meng, D. Rubin, Maximum likelihood estimation via the ECM algorithm: a general framework, Biometrika 80 (2) (1993) 267.
[57] W. Winkler, Using the EM algorithm for weight computation in the Fellegi-Sunter model of record linkage, in: Survey Research Methods, American Statistical Association, vol. 667, 1988, p. 671.
[58] W. Winkler, Near automatic weight computation in the Fellegi-Sunter model of record linkage, in: Fifth Annual Research Conference, US Bureau of the Census, 1989.
[59] S. Grannis, J. Overhage, C. McDonald, Analysis of identifier performance using a deterministic linkage algorithm, in: AMIA Annual Symposium, 2002, p. 305.
[60] W. Cohen, Data integration using similarity joins and a word-based information representation language, ACM Transactions on Information Systems 18 (3) (2000) 288–321.
[61] M.G. Elfeky, V.S. Verykios, A.K. Elmagarmid, TAILOR: a record linkage toolbox, in: IEEE ICDE, San Jose, 2002, pp. 17–28.
[62] M. Bilenko, R.J. Mooney, Adaptive duplicate detection using learnable string similarity measures, in: ACM SIGKDD, Washington, DC, 2003, pp. 39–48.
[63] B. On, N. Koudas, D. Lee, D. Srivastava, Group linkage, in: IEEE ICDE, Istanbul, Turkey, 2007, pp. 496–505.
[64] M. Herschel, F. Naumann, S. Szott, M. Taubert, Scalable iterative graph duplicate detection, IEEE Transactions on Knowledge and Data Engineering 24 (11) (2012) 2094–2108.
[65] I.H. Witten, A. Moffat, T.C. Bell, Managing Gigabytes, 2nd edition, Morgan Kaufmann, 1999.
[66] V. Raghavan, P. Bollmann, G. Jung, A critical investigation of recall and precision as measures of retrieval system performance, ACM Transactions on Information Systems 7 (3) (1989) 205–229.
[67] C. Manning, H. Schütze, MITCogNet, Foundations of Statistical Natural Language Processing, vol. 59, MIT Press, 1999.
[68] N. Lavrač, P. Flach, B. Zupan, Rule evaluation measures: a unifying view, Inductive Logic Programming (1999) 174–185.
[69] V.S. Verykios, A. Karakasidis, V. Mitrogiannis, Privacy preserving record linkage approaches, International Journal of Data Mining, Modelling and Management 1 (2) (2009) 206–221.
[70] R. Hall, S. Fienberg, Privacy-preserving record linkage, in: Privacy in Statistical Databases, Springer Lecture Notes in Computer Science, vol. 6344, Corfu, Greece, 2010, pp. 269–283.
[71] T. Churches, P. Christen, Some methods for blindfolded record linkage, BioMed Central Medical Informatics and Decision Making 4 (9) (2004).
[72] A. Al-Lawati, D. Lee, P. McDaniel, Blocking-aware private record linkage, in: International Workshop on Information Quality in Information Systems, Baltimore, MD, USA, 2005, pp. 59–68.
[73] T. Bachteler, R. Schnell, J. Reiher, An empirical comparison of approaches to approximate string matching in private record linkage, in: Statistics Canada Symposium, 2010.
[74] A. Inan, M. Kantarcioglu, E. Bertino, M. Scannapieco, A hybrid approach to private record linkage, in: IEEE ICDE, Cancun, Mexico, 2008, pp. 496–505.
[75] M. Yakout, M. Atallah, A. Elmagarmid, Efficient private record linkage, in: IEEE ICDE, Shanghai, 2009, pp. 1283–1286.
[76] S. Weber, H. Lowe, A. Das, T. Ferris, A simple heuristic for blindfolded record linkage, Journal of the American Medical Informatics Association 19 (e1) (2012) e157–e161.
[77] C. O'Keefe, M. Yung, L. Gu, R. Baxter, Privacy-preserving data linkage protocols, in: ACM Workshop on Privacy in the Electronic Society, Washington, DC, USA, 2004, pp. 94–102.
[78] E. Durham, Y. Xue, M. Kantarcioglu, B. Malin, Quantifying the correctness, computational complexity, and security of privacy-preserving string comparators for record linkage, Information Fusion 13 (4) (2012) 245–259.
[79] A. Karakasidis, V.S. Verykios, Advances in privacy preserving record linkage, in: E-activity and Innovative Technology, Advances in Applied Intelligence Technologies Book Series, IGI Global, 2010, pp. 22–34.
[80] S. Trepetin, Privacy-preserving string comparisons in record linkage systems: a review, Information Security Journal: A Global Perspective 17 (5) (2008) 253–266.
[81] P. Christen, Geocode matching and privacy preservation, in: KDD Workshop on Privacy, Security, and Trust, Las Vegas, 2009, pp. 7–24.
[82] O. Goldreich, Foundations of Cryptography: Basic Applications, vol. 2, Cambridge University Press, 2004.
[83] Y. Lindell, B. Pinkas, Secure multiparty computation for privacy-preserving data mining, Journal of Privacy and Confidentiality 1 (1) (2009) 5.
[84] Y. Lindell, B. Pinkas, An efficient protocol for secure two-party computation in the presence of malicious adversaries, EUROCRYPT (2007) 52–78.
[85] R. Canetti, Security and composition of multiparty cryptographic protocols, Journal of Cryptology 13 (1) (2000) 143–202.
[86] L. Dusserre, C. Quantin, H. Bouzelat, A one way public key cryptosystem for the linkage of nominal files in epidemiological studies, Medinfo 8 (1995) 644–647.
[87] C. Quantin, H. Bouzelat, F. Allaert, A. Benhamiche, J. Faivre, L. Dusserre, How to ensure data security of an epidemiological follow-up: quality assessment of an anonymous record linkage procedure, International Journal of Medical Informatics 49 (1) (1998) 117–122.
[88] H. Bouzelat, C. Quantin, L. Dusserre, Extraction and anonymity protocol of medical file, in: AMIA Fall Symposium, 1996, pp. 323–327.
[89] C. Quantin, H. Bouzelat, F.-A. Allaert, A.-M. Benhamiche, J. Faivre, L. Dusserre, Automatic record hash coding and linkage for epidemiological follow-up data confidentiality, Methods of Information in Medicine 37 (3) (1998) 271–277.
[90] B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd edition, John Wiley & Sons Inc., New York, 1996.
[91] H. Krawczyk, M. Bellare, R. Canetti, HMAC: Keyed-Hashing for Message Authentication, RFC Editor, 1997.
[92] S. Singh, The Code Book: The Secret History of Codes and Code-breaking, Fourth Estate-E-books-General, 2010.
[93] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, M. Zhu, Tools for privacy preserving distributed data mining, SIGKDD Explorations 4 (2) (2002) 28–34.
[94] A. Yao, How to generate and exchange secrets, in: 27th Annual Symposium on Foundations of Computer Science, IEEE, 1986, pp. 162–167.
[95] O. Goldreich, S. Micali, A. Wigderson, How to play any mental game, in: 19th Annual ACM Symposium on Theory of Computing, New York, USA, 1987, pp. 218–229.
[96] R. Agrawal, A. Evfimievski, R. Srikant, Information sharing across private databases, in: ACM SIGMOD, San Diego, 2003, pp. 86–97.
[97] L. Kissner, D. Song, Private and Threshold Set-intersection, Technical Report, Carnegie Mellon University, 2005.
[98] M. Luby, C. Rackoff, How to construct pseudo-random permutations from pseudo-random functions, in: CRYPTO, vol. 85, 1986, p. 447.
[99] D. Song, D. Wagner, A. Perrig, Practical techniques for searches on encrypted data, in: IEEE Symposium on Security and Privacy, 2000, pp. 44–55.
[100] M. Freedman, Y. Ishai, B. Pinkas, O. Reingold, Keyword search and oblivious pseudorandom functions, Theory of Cryptography (2005) 303–324.
[101] A. Karakasidis, V.S. Verykios, P. Christen, Fake injection strategies for private phonetic matching, in: International Workshop on Data Privacy Management, Leuven, Belgium, 2011.
[102] M. Scannapieco, I. Figotin, E. Bertino, A. Elmagarmid, Privacy preserving schema and data matching, in: ACM SIGMOD, Beijing, China, 2007, pp. 653–664.
[103] C. Pang, L. Gu, D. Hansen, A. Maeder, Privacy-preserving fuzzy matching using a public reference table, Intelligent Patient Management (2009) 71–89.
[104] D. Vatsalan, P. Christen, V.S. Verykios, An efficient two-party protocol for approximate matching in private record linkage, in: AusDM, CRPIT, vol. 121, Ballarat, Australia, 2011, pp. 125–136.
[105] L. Sweeney, K-anonymity: a model for protecting privacy, International Journal of Uncertainty Fuzziness and Knowledge Based Systems 10 (5) (2002) 557–570.

[106] N. Li, T. Li, S. Venkatasubramanian, T-closeness: privacy beyond k-anonymity and l-diversity, in: IEEE ICDE, Istanbul, Turkey, 2007, pp. 106–115.
[107] A. Machanavajjhala, D. Kifer, J. Gehrke, M. Venkitasubramaniam, l-diversity: privacy beyond k-anonymity, ACM Transactions on Knowledge Discovery from Data 1 (1) (2007) 3.
[108] M. Kantarcioglu, W. Jiang, B. Malin, A privacy-preserving framework for integrating person-specific databases, in: Privacy in Statistical Databases, Springer, 2008, pp. 298–314.
[109] N. Mohammed, B. Fung, M. Debbabi, Anonymity meets game theory: secure data integration with malicious participants, International Journal on Very Large Databases 20 (4) (2011) 567–588.
[110] A. Karakasidis, V.S. Verykios, Reference table based k-anonymous private blocking, in: 27th Annual ACM Symposium on Applied Computing, Trento, Italy, 2012.
[111] B. Bloom, Space/time trade-offs in hash coding with allowable errors, Communications of the ACM 13 (7) (1970) 422–426.
[112] A. Broder, M. Mitzenmacher, Network applications of bloom filters: a survey, Internet Mathematics 1 (4) (2004) 485–509.
[113] E. Durham, Y. Xue, M. Kantarcioglu, B. Malin, Private medical record linkage with approximate matching, in: AMIA Annual Symposium, Washington, DC, 2010, p. 182.
[114] R. Schnell, T. Bachteler, J. Reiher, Privacy-preserving record linkage using bloom filters, BMC Medical Informatics and Decision Making 9 (1) (2009).
[115] A. Karakasidis, V.S. Verykios, Secure blocking + secure matching = secure record linkage, Journal of Computing Science and Engineering 5 (2011) 223–235.
[116] P. Lai, S. Yiu, K. Chow, C. Chong, L. Hui, An efficient Bloom filter based solution for multiparty private matching, in: SAM, Las Vegas, 2006, p. 7.
[117] D. Vatsalan, P. Christen, An iterative two-party protocol for scalable privacy-preserving record linkage, in: AusDM, CRPIT, vol. 134, Sydney, Australia, 2012.
[118] H. Kargupta, S. Datta, Q. Wang, K. Sivakumar, On the privacy preserving properties of random data perturbation techniques, in: IEEE ICDM, Florida, USA, 2003, pp. 99–106.
[119] A. Inan, M. Kantarcioglu, G. Ghinita, E. Bertino, Private record matching using differential privacy, in: EDBT, Lausanne, Switzerland, 2010, pp. 123–134.
[120] C. Dwork, Differential privacy, in: International Colloquium on Automata, Languages and Programming, 2006, pp. 1–12.
[121] C. Papadimitriou, Computational Complexity, John Wiley and Sons Ltd., 2003.
[122] M. Kuzu, M. Kantarcioglu, E. Durham, B. Malin, A constraint satisfaction cryptanalysis of Bloom filters in private record linkage, in: Privacy Enhancing Technologies, Springer, 2011, pp. 226–245.
[123] C. Shannon, W. Weaver, The Mathematical Theory of Communication, vol. 19, University of Illinois Press, Urbana, 1962.
[124] P. Christen, A. Pudjijono, Accurate synthetic generation of realistic personal information, in: PAKDD, Lecture Notes in Artificial Intelligence, vol. 5476, Springer, Bangkok, Thailand, 2009, pp. 507–514.
[125] J. Hoag, C. Thompson, A parallel general-purpose synthetic data generator, ACM SIGMOD 36 (1) (2007) 19–24.
[126] E. Van Eycken, K. Haustermans, F. Buntinx, A. Ceuppens, J. Weyler, E. Wauters, O. Van, et al., Evaluation of the encryption procedure and record linkage in the Belgian National Cancer Registry, Archives of Public Health 58 (6) (2000) 281–294.
[127] W. Du, M. Atallah, Protocols for secure remote database access with approximate matching, in: E-Commerce Security and Privacy, Springer, 2001.
[128] E. Durham, A Framework for Accurate, Efficient Private Record Linkage, Ph.D. Thesis, Vanderbilt University, 2012.
[129] M. Atallah, F. Kerschbaum, W. Du, Secure and private sequence comparisons, in: Workshop on Privacy in the Electronic Society, ACM, Washington, DC, USA, 2003.
[130] P. Ravikumar, W. Cohen, S. Fienberg, A secure protocol for computing string distance metrics, in: Workshop on Privacy and Security Aspects of Data Mining held at IEEE ICDM, Brighton, UK, 2004.
[131] F. Li, Y. Chen, B. Luo, D. Lee, P. Liu, Privacy preserving group linkage, in: Scientific and Statistical Database Management, Springer, 2011, pp. 432–450.
[132] G. Hjaltason, H. Samet, Properties of embedding methods for similarity searching in metric spaces, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (5) (2003) 530–549.
[133] D. Lin, An information-theoretic definition of similarity, in: ICML, vol. 1, Madison, WI, USA, 1998, pp. 296–304.
[134] B. Hawashin, F. Fotouhi, T. Truta, A privacy preserving efficient protocol for semantic similarity join using long string attributes, in: ACM Workshop on Privacy and Anonymity in the Information Society, Sweden, 2011, p. 6.
[135] R. Coifman, S. Lafon, Diffusion maps, Applied and Computational Harmonic Analysis 21 (1) (2006) 5–30.
[136] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, R. Harshman, Indexing by latent semantic analysis, Journal of the American Society for Information Science 41 (6) (1990) 391–407.
[137] X. He, P. Niyogi, Locality Preserving Projections, vol. 103, MIT Press, Vancouver, Canada, 2004.
[138] W. Jiang, C. Clifton, A secure distributed framework for achieving k-anonymity, International Journal on Very Large Data Bases 15 (4) (2006) 316–333.
[139] M. Sadinle, R. Hall, S. Fienberg, Approaches to multiple record linkage, in: International Statistical Institute, Dublin, Ireland, 2011.
[140] L. Wu, X. Ying, X. Wu, Reconstruction from randomized graph via low rank approximation, in: SIAM, Columbus, Ohio, USA, 2010, pp. 60–71.
[141] E. Zheleva, L. Getoor, Preserving the privacy of sensitive relationships in graph data, in: 1st ACM SIGKDD Workshop on Privacy, Security, and Trust, Springer-Verlag, San Jose, USA, 2007, pp. 153–171.
[142] M. Reiter, A. Rubin, Crowds: anonymity for web transactions, ACM Transactions on Information and System Security 1 (1) (1998) 66–92.
[143] D. Barone, A. Maurino, F. Stella, C. Batini, A privacy-preserving framework for accuracy and completeness quality assessment, Emerging Paradigms in Informatics, Systems and Communication (2009) 83.