Learning from Encoded Patterns: A Recurrence Plot Approach for Privacy-Preserving Record Linkage

Thiago Nóbrega[*]
thiagonobrega@dsc.ufcg.edu.br

Tiago Brasileiro Araújo[§ †]
tiago.brasileiro@ifpb.edu.br

Dimas Cassimiro[‡]
dimas.cassimiro@ufape.edu.br

Demetrio Mestre[*]
demetrio@dsc.ufcg.edu.br

Carlos Eduardo S. Pires[*]
cesp@dsc.ufcg.edu.br

Kostas Stefanidis[†*]
konstantinos.stefanidis@tuni.fi

Abstract

Privacy-Preserving Record Linkage (PPRL) enables the integration of records referring to the same entities across multiple datasets without disclosing sensitive information. Traditional PPRL techniques often rely on predefined similarity metrics that may not capture complex data relationships (e.g., non-linear relationships), limiting linkage quality. This paper presents a Cross Recurrence Plot (CRP)–based representation of Bloom Filter pairs to train machine learning and deep learning classifiers capable of distinguishing matching and non-matching records directly from encoded bit patterns. The proposed workflow simplifies the PPRL process by replacing similarity thresholding with an ML-based decision model while maintaining privacy guarantees. Experimental results using real-world datasets demonstrate that the proposed approach outperforms traditional threshold-based methods in terms of linkage quality, confirming its potential to support trusted and privacy-aware data sharing.

CCS Concepts

• **Security and privacy** → **Data anonymization and sanitization**; *Security services*; • **Information systems** → *Data management systems.*

Keywords

Entity resolution, Privacy preserving entity resolution, Data privacy, Bloom Filter, Recurrence plot representation

1 Introduction

Data has become a critical asset across scientific, governmental, and industrial domains, supporting activities that range from evidence-based public policy to large-scale business analytics [4]. Governments, for example, routinely integrate demographic and medical information to guide healthcare decisions [2, 16, 18], while companies combine heterogeneous customer datasets to validate records, improve recommendations, and streamline on-boarding processes [12, 14]. Because these practices involve sensitive personal information, they must comply with legal frameworks such as the GDPR and HIPAA, which demand rigorous privacy protection. In this setting, Privacy-Preserving Record Linkage (PPRL) has emerged as a key technology, enabling the identification of entities across datasets without exposing unique identifiers. PPRL achieves this goal by transforming quasi-identifiers (QIDs) into anonymized or encoded representations and then comparing these representations to decide whether (or not) records from different data sources refer to the same entity [4].

Despite its promise, PPRL remains a technically demanding task. Most existing solutions classify record pairs using a threshold-based strategy: the parties involved select a similarity threshold—typically between 0 and 1—and declare two records a match if their similarity exceeds this value. However, setting this threshold is far from trivial. A value that is too high (e.g., 0.9) may cause the system to miss potential true matches, while a value that is too low increases the risk of false positives. Determining the *right* threshold often requires expert intervention and extensive parameter tuning, and even small deviations can lead to significant losses in linkage quality.

To overcome the limitations of manual threshold selection, researchers have investigated Machine-Learning (ML) methods that treat PPRL as a classification problem [10]. Typical approaches employ statistical models, such as support vector machines or logistic regression, trained on labelled record-pair similarity scores. Although ML models can reduce the dependence on a single threshold, they remain constrained by the nature of the similarity measures themselves. Similarity functions, such as Jaccard and Dice, are typically selected heuristically and highly task-dependent [8]. In the PPRL context, they operate over binary encodings (e.g., Bloom filters), which capture only limited syntactic similarity [4]. As a result, the predictive power of these ML models is fundamentally limited by the quality of the similarity scores they receive as input.

In contrast, in traditional Record Linkage (RL) settings where raw data are available, deep learning (DL) has been successfully applied to learn complex, non-linear relationships directly from the data. Neural networks, autoencoders, and other DL architectures can capture semantic patterns that go well beyond handcrafted similarity functions, leading to significant improvements in linkage quality [1, 13, 5]. These improvements strongly suggest that PPRL could also benefit from DL methods, especially if an effective way can be found to represent the privacy-preserving encoded data in a form that neural networks can exploit.

[*] ∗ Federal University of Campina Grande, Brazil
[†] Data Science Research Centre, Tampere University, Finland
[‡] Federal University of Agreste de Pernambuco, Brazil
[§] Federal Institute of Paraíba, Brazil

SAC'26, Thessaloniki, Greece

ACM ISBN 979-X-XXXX-XXXX-X/26/03
https://doi.org/10.1145/3748522.3779921

However, directly applying DL to PPRL is challenging. The encoding process itself acts as a dimensionality reduction with noise addition. For example, transforming a 128-bit UTF-8 representation of a six-letter word into a 12-bit Bloom filter inevitably discards syntactic and semantic information. Moreover, hash collisions and hardening techniques introduce additional randomness. From the classifier's perspective, these operations create a noisy, low-information space that makes it difficult to detect meaningful patterns. For instance, consider the comparison of two semantically related sentences, such as "The mayor removed the bus stop" and "The city authority moved the bus terminal". While a human reader can easily recognize the shared meaning of mayor and city authority, the PPRL process may reduce these sentences to bit arrays whose direct similarity might be only 0.5, leaving a classifier with little evidence of their relationship.

To address these challenges, we introduce a Cross Recurrence Plot (CRP)–based representation of PPRL data. Originally developed to analyze complex dynamical systems [9], RPs transform time-series or high-dimensional signals into two-dimensional images that reveal latent structures such as periodicity, self-similarity, and hidden correlations. Our key insight is that PPRL encodings, such as Bloom filters, can also be interpreted as signals whose internal patterns may reflect the underlying relationships between records. By mapping encoded record pairs into RP images, we provide a richer and more expressive representation that allows deep neural networks to learn directly from the encoded data themselves, bypassing the need for predefined similarity metrics.

This research advances privacy-preserving mechanisms that ensure secure and reliable data integration across organizational boundaries, thereby fostering trust and enhancing interoperability in modern data-sharing ecosystems. Our approach strengthens the foundations for collaborative analytics while ensuring that data confidentiality and linkage quality remain balanced. In this light, we make the following contributions:

- **Novel representation:** We propose a new RP-based method for representing pairs of PPRL-encoded records, enabling classifiers to exploit structural patterns that are invisible to traditional similarity measures;
- **Comprehensive evaluation:** We conduct extensive experiments on real-world datasets using different encoding parameters and privacy hardening techniques. Our evaluation compares the proposed approach with state-of-the-art PPRL classification and hardening methods, demonstrating improvements in linkage quality across diverse settings.

The rest of the paper is structured as follows. Section 2 reviews related work on PPRL and learning-based approaches. Section 3 outlines the main concepts. Section 4 introduces the proposed CRP-based representation for PPRL classifiers, while Section 5 details the CRP-based workflow. Section 6 presents the experimental setup and results, and Section 7 concludes the paper with final remarks and future directions.

2 Related Work

Record Linkage (RL) has benefited from machine learning techniques to improve the classification of record pairs [6, 4]. Many of these approaches are based on supervised learning and assume the availability of labeled training data—a condition rarely met in real-world applications. In the context of PPRL, this challenge is amplified, as privacy constraints make it infeasible to manually label datasets containing sensitive information [18].

To address the scarcity of labeled data, recent work has explored the use of Transfer Learning (TL). Nóbrega et al. [11] propose AT-UC, a framework based on unsupervised domain adaptation. Their approach leverages knowledge from a labeled, public source dataset to train a classifier for an unlabeled, private target dataset. The core idea is to select a suitable source dataset that is most related to the target, thereby enabling the training of an effective decision model without accessing or disclosing any sensitive information from the target domain. This method focuses on adapting the classification model, assuming the underlying data representation is fixed.

Christen et al. [5] introduced a novel technique using autoencoders to transform BFs into lower-dimensional, dense numerical vectors. This transformation effectively masks the original bit patterns, making it hard to execute frequency-based attacks, while providing good classification results.

Ranbaduge et al. [13] proposed a deep learning-based PPRL protocol for multi-party scenarios. The protocol employs autoencoders, allowing each data owner to generate distinct encodings. As these representations are not directly comparable, a linkage unit is introduced to learn a mapping function that projects the autoencoder-generated vectors into a common vector space, thereby enabling accurate similarity comparisons while preserving privacy.

As an alternative, other works propose entirely different encoding schemes. Ziyad et al. [20] developed the Reference Set-based Encoding (RSE) method, which generates bit arrays of a fixed length and with a constant number of 1-bit for every record. This design directly counters length and frequency-based attacks by making all encodings structurally uniform. The method offers a transparent alternative to BF encoding, with a clear trade-off between linkage quality and privacy.

While these approaches represent advances in either adapting classifiers or securing data representations, the exploration of alternative encodings that are inherently secure and efficient remains open. Our work contributes to this latter direction by proposing a novel representation designed to enable new comparison methods in PPRL.

3 Building Blocks

This section introduces the core concepts underlying the topics discussed in this paper.

3.1 Bloom Filters

A BF consists of a vector of l-bits (filter length), with all bits set to '*0*', initially. The BF can be formalized as $[b_0, \cdots, b_l]$, where b_m represents the bit of position m. To insert a set of elements ($S = \{s_1, \cdots, s_n\}$) in a BF, k independent *hash functions*[1], $H(x) = \{h_1(x), \cdots, h_k(x)\}$, are employed to map the elements $s_i \in S$ to the l-bits vector. Furthermore, the output of $H(x)$ indicates the bits (b) that need to be set to '*1*' in the BF.

[1] A hash function is an algorithm that takes messages and maps them to a value of a certain length, called a hash value or hash.

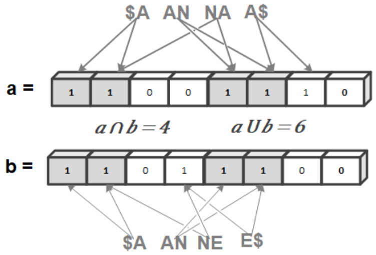


Figure 1: Inserting the names ANA and ANE into 8-bit Bloom filters (l=8 and k=2).

The quality of the anonymization depends on the BF parametrization, number of hash functions (k), and filter length (l) [17]. For a given number of elements (n) to be inserted into the BF, the probability of a specific bit still '*0*' is $p = e^{-\frac{k \times n}{l}}$. We can choose a k to minimize the probability of two different elements being mapped to the same bit position (f) by setting $p = 0.5$ [3]. In other words, the probability of a bit in the BF still 0 (or flipped to 1) should be 0.5 to reduce the false-positive rate. For PPRL, this is relevant because the bit patterns and their frequencies in a set of BF can be exploited by frequency and cryptanalysis [19] attacks. Such attacks exploit the fact that BFs that are almost empty can provide information about rare elements. As a result, the re-identification of the entities is facilitated [17].

Figure 1 illustrates the insertion of the names **ANA** and **ANE** into the BFs a and b, respectively. First, each name is transformed into bigrams; then, each bigram is mapped by a hash function into a BF position. Finally, the positions are changed to '*1*' in the BF.

The BF technique also enables the similarity calculation of two filters (a and b) through token distance functions, such as $Jaccard = \frac{|a \cap b|}{|a \cup b|}$, where $|a \cap b|$ is the number of positions with the value *1* that coincide in both filters; and $|a \cup b|$ represents the number of positions with the value *1* in the union of the filters a and b. Regarding the example illustrated in Figure 1, the $Jaccard$ similarity between filter a (representing the name ANA) and filter **b** (ANE) is equal to $\frac{4}{6} = 0.66$.

In summary, the BF is relatively easy to understand and implement compared to other more complex techniques. Also, the computation over the BF bit-vectors is efficient and provides a good enough accuracy of the calculated similarity. Therefore, given these advantages, the BF is considered the anonymization method of choice for PPRL.

3.2 Recurrence in dynamical system

Dynamical systems theory is used to describe the behavior of complex dynamical systems, from complex meteorological cycles (e.g., El/Niño) to problems in medical and economic contexts [9]. An important task in understanding dynamical systems is the discovery of the recurrences of system states, e.g., the periodicity of meteorological events. Recurrence Plot (RP) is a tool that can be used to investigate recurrences in a dynamical system.

The concept behind the RP is to represent the time when states x_i recur in the system. In summary, RP enables the analysis of an m-dimensional space through a two-dimensional representation of its recurrences [9]. Such recurrence of a state at time i at a different time j is pictured within a two-dimensional square matrix R. Equation 1 formalizes the RP.

$$R^{m,\varepsilon}_{i,j} = \Theta(\varepsilon - ||x_i - x_j||), i, j = 1, \cdots, N \tag{1}$$

where ε is threshold distance, $\Theta(\cdot)$ the Heaviside function[2], $|| \cdot ||$ norm (L_1,L_2 or L_∞) and m represents the m-dimensional neighborhood of size.

In order to study the correlations (also named synchronization) between two dynamical systems, several bivariate recurrence techniques were proposed. The Cross Recurrence Plot (CRP) is a bivariate recurrence technique that was proposed to investigate (simultaneously) two different dynamical system evolution, allowing the study of dependencies between two different systems [9]. In other words, the CRP observes the states of both systems, and if the state recurs in both, the CRP will capture it. Suppose we have two dynamical systems, each one represented by x_i and y_i in the same d-dimensional space. We find the corresponding cross recurrence matrix by computing the pairwise mutual distances between the vectors of the two systems. Equation 2 represents the CRP calculation:

$$CR^{x,y}_{i,j}(\varepsilon) = \Theta(\varepsilon - ||x_i - y_j||), i, j = 1, \cdots, N \tag{2}$$

Next, we explain how to apply CRP to Bloom Filters. We aim to uncover latent patterns of alignment between filters that are not easily captured by standard similarity measures, thereby enhancing the reliability of linkage while preserving privacy.

4 Novel Data Representation for PPRL Classifiers

This section proposes an alternative representation of Bloom Filter pairs that enables a novel comparison method in the PPRL context. First, we detail how to represent BF pairs as RP. Then, in the subsequent section, to validate the use of RP in the PPRL context, we propose a methodology to incorporate RP in a PPRL workflow.

We hypothesize that representing BF pairs as RPs within a dynamical system framework enables more effective classification strategies for PPRL. Rather than relying solely on similarity scores between encoded record pairs, our approach leverages the structural properties of RPs to train a supervised classifier. This classifier is then used to perform the linkage task.

4.1 Recurrence Plot in PPRL classifiers

The use of RP to represent pairs of encoded records intends to replace the standard similarity measures (e.g., Jaccard and Dice) by the RP. The RP aims to highlight patterns that will be employed as features to the PPRL Classification step. Moreover, the use of RP to represent record pairs will allow ML techniques to learn from the encoded records, reducing the bias introduced by the standard similarity measures [8]. In summary, RP will enable a more precise classifiers (i.e., Neural networks) in the PPRL context.

Before explaining how to employ RP within the PPRL context, first, we need to present assumptions regarding the PPRL encoded data. In our experiments and the formalization presented in this section, we consider a 1-dimensional array of bits, such as the

[2] The Heaviside function is commonly used in control theory and signal processing to represent a signal that switches on at a specified time [9]

Bloom Filter technique. Thus, each position of the encoded data array ($\hat{e}$) represents a state that needs to be mapped by RP, such as: $\hat{e} = [x_1, \cdots, x_l]$, where x_i represents the value (bit) in the encoded data position whilst l is the length of the encoded data.

In a typical PPRL process, pairs of encoded data are used in the Comparison step. Thus, to highlight the states (encoded data patterns) that simultaneously occur in both records, we must operate a bivariate RP [9]. In this sense, we employed the Cross Recurrence Plot (CRP) as our bivariate technique because it can show which states in a dynamical system occur simultaneously in a second dynamical system. Consequently, the aforementioned CRP characteristic can be employed to detect regions of the encoded data (i.e., BF segments) that coincide.

CRP will be employed to identify similar regions of the encoded records. In other words, the CRP highlights common states (e.g., bits) over encoded record pairs. Moreover, these regions (the RPs) will be forwarded to a ML-based classifier to extract patterns that can identify matching records.

In Section 3.2, we presented a generic CRP formalization (Equation 2). In order to provide further details of the CRP in the PPRL context, we introduce a modified version of CRP. Equation 3 formalizes CRP using two encoded records ($\hat{e}_1$ and $\hat{e}_2$) and m neighbors. The m-neighbors are used to create and delimitate an area around each bit position of the encoded record that the Heaviside function will compute. For a graphical example of the m-neighbors, see the red dashed areas of Figure 2.

$$CRP(\hat{e}_1, \hat{e}_2, m, \epsilon) = \sum_{i=0}^{l_1} \sum_{j=0}^{l_2} \Theta\left(\alpha, \sum_{w=0}^{m} ||\hat{e}_1[i+w] - \hat{e}_2[j+w]||\right) \quad (3)$$

In Eq. 3, l_1 and l_2 are the lengths of the encoded records, α is a threshold distance employed in the Heaviside function (Θ), and $||\hat{e}_1[i] - \hat{e}_2[j]||$ is the distance between two sets of elements of the encoded records $\hat{e}_1[i]$ and $\hat{e}_2[j]$.

Notice that CRP will generate a 2-dimensional matrix, $CRP_{i,j}$. In a typical PPRL encoding process, all parties encode their data using the same number of bits (l). Thus, we can calculate the $CRP_{n,n}$ dimensions using l and m. Equation 4 shows the formula.

$$n = l - (m-1) \quad (4)$$

As disclosed in Equation 3, CRP will iterate over the elements (e.g., bits) of the encoded records and compute the distance of these elements considering m neighbors. Later, this distance will be compared against a threshold (α) in a Heaviside function. In Equation 5, we define the Heaviside function used in our solution.

$$\Theta(\alpha_i, v) = \begin{cases} 1: & v \leq \alpha_i \\ 0: & v > \alpha_i \end{cases} \quad (5)$$

To illustrate the use of CRP over encoded data, consider the encoded records of ANA and ANE (depicted in Section 3.1). In Figure 2, we illustrate the CRP encoding process for the BF ($l = 8$) of ANA and ANE. In the example, we consider three neighbors ($m = 3$) and the Heaviside threshold equals to one ($\alpha = 1$).

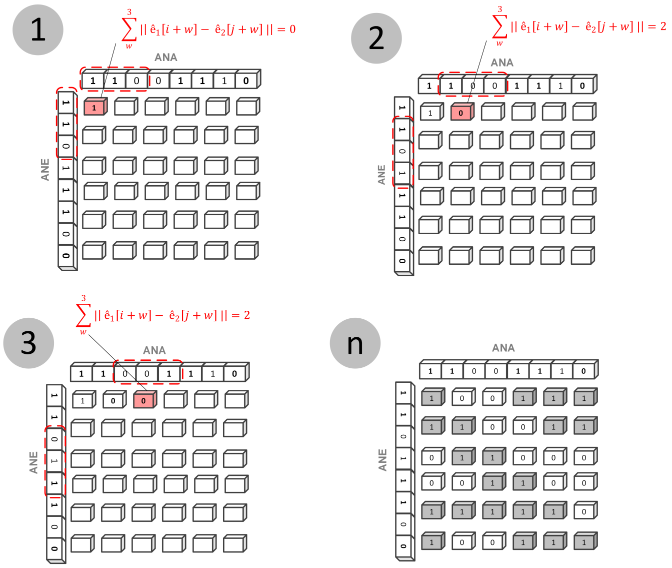

1
$\sum_{w}^{3} || \hat{e}_1[i+w] - \hat{e}_2[j+w] || = 0$
2
$\sum_{w}^{3} || \hat{e}_1[i+w] - \hat{e}_2[j+w] || = 2$
3
$\sum_{w}^{3} || \hat{e}_1[i+w] - \hat{e}_2[j+w] || = 2$
n
ANA
ANE
1 1 0 0 1 1 1 0
1 1 0 1 1 1 0 0

Figure 2: CRP encoding process

The CRP is represented as a 6 × 6 matrix. This dimension is defined by the filter length and the number of neighbors (Equation 4, $n = 8 - (3-1)$). Each element in the matrix is calculated using Equation 3. The Heaviside function ($\Theta(\alpha, \sum_{w=0}^{m} ||\hat{e}_1[i+w] - \hat{e}_2[j+w]||)$) in the last segment of Equation 3 is employed to define whether a filter position represents a co-occurrence in both encoded records, or not. In other words, the first segment of the equation ($\sum_{i=0}^{l_1} \sum_{j=0}^{l_2}$) iterates over each position of the $CRP_{i,j}$, while the second one (the Heaviside function) marks the position with 1 or 0 if a region of both filters indicates a co-occurrence (or not), respectively.

The first step of Figure 2 illustrates the computation of the element $CRP_{0,0}$. The red square illustrates the neighborhood (m) used to define whether a bit position co-occurs in both filters. Notice that the distance between the regions is calculated as $\sum_{w=0}^{m} ||\hat{e}_1[i+w] - \hat{e}_2[j+w]||$. This information is highlighted in red. The calculated distance (Manhattan distance) is zero for the first step because the regions are identical. This distance is compared against the Heaviside threshold ($\alpha = 1$) and the region is marked as co-occurrence in the CRP.

The process is repeated for every element of the CRP matrix, and the position is marked accordingly. For instance, in the second step, the distance (2) is greater than α. Therefore, the Heaviside function marks the region as zero. Finally, at the end of the CRP generation, the matrix will be filled with zeros and ones, evidencing similar regions in the encoded data.

By representing the encoded record pair as a CRP, we expect to highlight the local similarity/difference of the data. Moreover, following the literature on Recurrence Quantification [9], a CRP of two systems (encoded data in our context) is represented as a series of parallel segments in the matrix. The frequency and length of these lines are related to a certain similarity between both encoded data. Thus, by employing CRP over PPRL encoded data, we expect similar encoded data (e.g., matching records) to generate a particular

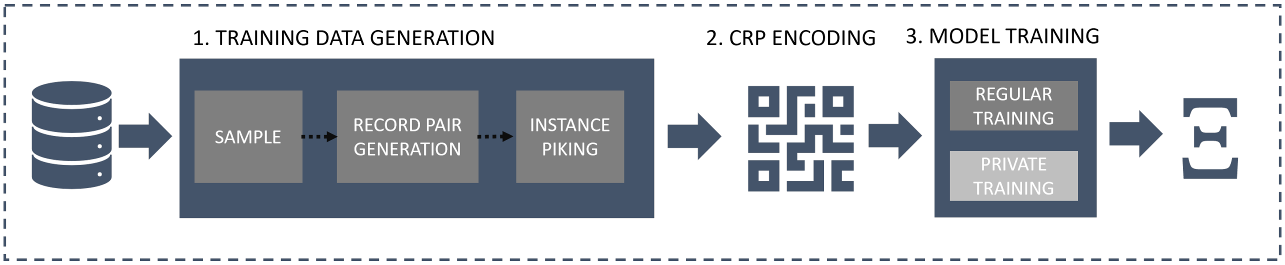
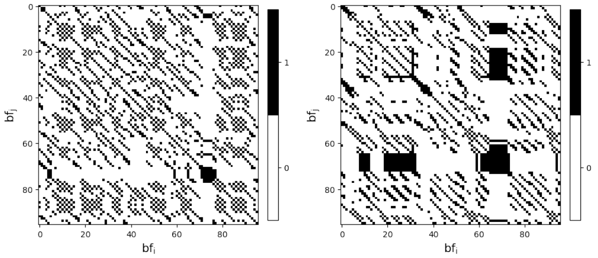


Figure 3: Proposed Workflow for Incorporating CRPs into the PPRL Process.

pattern (e.g., several bold and continuous diagonal lines). Figure 4 shows examples of matching and non-matching Bloom Filters pairs encoded with our CRP.



(a) Matching (0.9 of similarity) **(b) Non-matching (0.7 of similarity)**

Figure 4: CRP representation of two encoded record pairs. The records were encoded considering $l = 100$, $k = 5$, and a false positive rate of 0.5.

In Figure 4, although the Jaccard similarity values of 0.9 and 0.7 suggest a relatively small numerical difference between the two BF pairs, the visual patterns revealed through their RP representations are markedly distinct. This contrast underscores the potential of RP-based modeling to enhance classification performance in PPRL, by capturing structural nuances that traditional similarity metrics may overlook.

Notice that the matching encoded data exhibits the expected pattern, characterized by prominent diagonal structures. In contrast, the non-matching encoded data displays a reduced presence of diagonal lines and a higher concentration of black squares, indicating greater randomness and lower structural coherence. Several quantitative methods—analogous to binary similarity metrics—have been proposed to analyze such patterns in RPs [9]. Specifically, RP representations can be evaluated using: (i) frequency distribution, average length, longest length, and entropy of diagonal, vertical, and white vertical lines; (ii) pattern recurrence rate; (iii) determinism; (iv) divergence; (v) laminarity; and (vi) the laminarity-determinism ratio.

In this work, instead of using the previously mentioned numeric methods, CRP will be classified by ML-based classifiers. These classifiers will use the CRPs as input to the ML models. Instead of using a similarity value, or a small set of similarity values [10], as features, we will employ a 2-dimensional matrix of features, where the classifiers will look for partners that indicate matching encoded records. Moreover, instead of finding a decision boundary (line or hyperplane) based solely on similarity values, our work employs CRP as input to extract patterns representing matching records.

5 CRP-based Workflow for PPRL Classification

This section introduces a workflow that integrates the CRP approach into the PPRL classification step. Before describing the workflow in detail, we outline the assumptions and constraints regarding its components and functionalities.

The workflow operates under the Semi-Honest (HBC) adversarial model. Additionally, we assume the presence of a Semi-Trusted Third Party (STTP), responsible for executing the classification step. All other steps of the workflow are performed by the PPRL parties (data owners). The only information disclosed to the STTP consists of the CRP encoding parameters and the trained classifier.

The CRP representation serves two purposes: (i) to generate labeled training instances used to construct classifiers, and (ii) to provide an expressive feature space for classifying unlabeled record pairs. By leveraging structural patterns embedded in CRPs, the workflow overcomes the limitations of baseline threshold-based approaches that rely exclusively on single similarity scores.

It is important to note, however, that the workflow adopts a non-optimal strategy for training data selection: labeled instances are derived exclusively from the dataset of a single party. While this design maintains simplicity and avoids additional communication costs between parties, it also introduces potential bias into the resulting models. Since the sampled training data does not capture the heterogeneity across all participants, classifiers may generalize poorly when applied to data from other sources. Despite this limitation, such a design allows this work to evaluate the impact of the CRP in the PPRL classification workflow and compatible with the semi-honest setting considered here.

The proposed workflow comprises three distinct phases: (i) *training data generation*, (ii) *CRP encoding*, and (iii) *model training*. Figure 3 provides a visual overview of the process. Upon completion, the trained model is transferred to the STTP for deployment and evaluation.

5.1 Training Data Generation

First, a subset of records is randomly sampled from the dataset of a single participant, with sampling controlled by Υ_s ($0 < \Upsilon_s < 1$). This sample is partitioned into two subsets, $\bar{D}_a$ and $\bar{D}_b$, with a controlled fraction of matches Υ_m. In the *instance harvesting* step, similarities (e.g., Jaccard) are computed between candidate pairs, and only the hardest non-matching examples—those with similarity scores above the harvesting threshold Υ_h—are retained for training. This reduces the number of training pairs while encouraging the classifier to learn robust decision boundaries.

As the training data are drawn solely from one participant, the generated instance set may not adequately represent other party datasets. Consequently, the trained model might inherit statistical bias from this partial view of the data distribution.

5.2 CRP Encoding

The selected pairs are then transformed into the CRP representation using the Heaviside threshold (α) and neighborhood parameter (m), both agreed upon by the PPRL parties. This produces structured binary matrices that capture inter-record patterns in a representation more expressive than scalar similarity measures.

5.3 Model Training

Finally, the classifier is trained on the encoded CRPs. Two training modes are supported: (i) standard training without additional protection, or (ii) privacy-preserving training using Differential Privacy (DP), parameterized by ϵ. DP introduces calibrated noise into the optimization process, reducing the dependency of model parameters on individual records and thereby limiting the potential for information leakage.

In most existing PPRL solutions, ML classifiers (e.g., Logistic Regression and Support Vector Machines) are employed without differential privacy measures [4]. Our workflow is compatible with these standard classifiers, but it is also designed to support more expressive models (e.g., deep learning and decision-tree-based classifiers [7]), which can be trained under DP. In our implementation, we use the work of Abdi et al. [1]. The empirical trade-off between privacy guarantees and classification quality using DP is analyzed in Section 6.

It is important to highlight that the training procedure itself introduces statistical bias, as the data stem exclusively from a single participant. Thus, even though the privacy risks of model disclosure are mitigated (and further reduced if DP is applied), the resulting classifiers may reflect only a partial view of the full data distribution.

5.4 Computational Complexity

Instance harvesting requires quadratic complexity in the sample size Υ_s, but in practice it operates on relatively small samples ($100 \leq \Upsilon_s \leq 10,000$), and thus does not represent the dominant cost. The CRP encoding step is more expensive: with asymptotic cost $O(n^2)$ in the encoding length l when $m \ll l$, and up to $O(n^3)$ when $m \simeq l$. Since encoding applies to every instance, efficient blocking techniques are still recommended to avoid unnecessary comparisons [4], although the empirical evaluation of these techniques is beyond the scope of this work.

5.5 Workflow Output

At the end of the workflow, each participating party produces a trained classifier Ξ alongside anonymized encoded records, which are sent to the STTP. Although multiple models may be provided to the STTP, in this work, we do not propose any mechanism for combining or aggregating these classifiers. The STTP, therefore, operates directly on the individual models it receives. In this sense, we introduce DLC (Deep Learning Classifiers), which leverages the CRP representation of BF pairs to enable ML models to identify matching and non-matching records directly from encoded bit structures.

6 Evaluation

In this section, we evaluate the DLC effectiveness, i.e., the linkage and classifier quality. To this end, we present a discussion regarding the experimental results to answer the following Research Questions (RQ): (1) Is CRP able to improve the classifier's effectiveness? (2) Is DLC able to improve the PPRL quality results compared to the baseline and the competitor (AT-UC)? (3) What is the cost in linkage quality incurred during the Privacy-Preserving Training step ($\epsilon > 0$) of the workflow?

The linkage quality were evaluated in terms of Precision, Recall, and F1 metrics. Next, we present considerations regarding the datasets, anonymization parameters, ML classifiers, baselines, and competitors employed in our experiments.

6.1 Experimental Design

To assess our contributions, we assume that the PPRL parties use a Bloom Filter as their data encoding/anonymization technique. Moreover, we employ four pairs of real-world datasets to answer the aforementioned research questions. Table 1 presents a summary of the dataset characteristics and their anonymization parameters.

Table 1: Statistics of datasets

	DATASET DETAILS				BF		
name	record pair	matching pairs	attribute number	missing values	k	l	n
mvr	1×10^8	100	4	1,772	7	250	36
nvr	1×10^8	100	5	352	4	200	28
dblp-acm	6×10^6	500	3	14	3	450	94
census	6.85×10^5	80	3	291	3	100	23

The MHT workflow can use different ML techniques, from deep learning to statistical learning algorithms (e.g., SVM and Gradient Boosting classifiers). We explored this characteristic and tested the CRP data representation as well as the MHT workflow, considering the CRP Convolution Neutral Network (CCN). Moreover, we compared the CCN against two distinct families of classifiers: Deep Learning and classical ML classifiers. Table 2 illustrates our experimental design.

We employ the CRP in our CCN and well-known **classical Machine Learning classifiers**: Support Vector Machine (SVM[3]) and Gradient Boosting Classifier (GBC[4]). Furthermore, we compare our approach against a **threshold-based classifier** (baseline) and a **state-of-the-art TL classifier (AT-UC)** [11] as a competitor.

We varied several parameters in the MHT workflow, i.e., the Training Data Generation, CRP, and Model training stages. To optimize the experimental design, we considered the same random sample length (Υ_s = 10% of the original dataset), percentage of matching examples (Υ_m = 10% of the original dataset), metric functions, and CRP distance function. In the Training Data Generation

[3] trained with RBF kernel

[4] trained with 100 estimators, max depth of five, and learning rate 0.1

Table 2: Experimental design

parameter	values
Training Data Generation	
Random Sample Length Υ_s	10%
Percentage of matching examples (Υ_m)	10%
Percentage of non-matching examples (Υ_h)	50% to 300%
CRP encoding	
Heaviside function threshold (α)	1, 5, 10
CRP neighbors (m)	5, 10, 20, 30, 40
CRP distance function	Manhattan
Model training	
Classifiers	CCN, SCN, SVM, GBC
Metric Functions	Precision and Recall
Privacy Budget (ϵ)	1 , 5 , 10

phase, we varied the percentage of non-matching examples (Υ_h) to investigate the influence of the ratio between the matching and non-matching examples in the final linkage quality.

We varied the threshold α and the number of BF neighbors (m), in the CRP encoding phase. The variation of parameters (α,m) is employed to investigate the existence of a relation between the anonymization parameters - Bloom Filter length (l) and hash functions (k) - and the CRP parameters. Regarding the CRP distance function (presented in Equation 2 and 3), we employed the Manhattan distance.

Our approach and all baseline ones were implemented in Python 3, and we ran all experiments on a Linux server with 128 TensorCores (TPU), 2.4 GHz CPUs, and 48 GB of RAM. Moreover, considering the random sample in the first stages of the MHT workflow, we executed the experimental design five times to mitigate the influence of the sample in our results. The programs and datasets are available at the authors' repository[5].

6.2 Results

In this section, we answer the Research Questions raised in Section 6. The results consider the experimental design shown in Table 2.

6.2.1 Is CRP able to improve the classifier's effectiveness? To assess the impact of different CRP configurations (defined by parameters α and m) on linkage quality, Table 3 summarizes the Bloom Filter length (l), the average number of bigrams per filter (n), the number of hash functions (k), and the CRP configuration that achieved the best results in our experiments.

Table 3: BF encoding vs. CRP parameters.

Dataset	l	k	n	best crp_conf
ncvr	200	4	28	5x30
mvr	250	7	36	10x40
census	100	3	23	5x20
dblp_acm	450	3	94	5x30

The correlation between anonymization and CRP parameters is evidenced when we used a CRP configuration that employs: i) a Heaviside function threshold (α) close to the number of hash functions ($\alpha \approx k$); and ii) the number of neighbors (m) near to the mean number of n-grams of the BF ($m \approx n$). This parameter configuration ($\alpha \approx k$ and $m \approx n$) leaded to the best linkage results in our experiments.

The linkage results of this parametrization can be partially explained by the near-uniform distribution of bits ($fpr = .5$) performed by the Bloom Filter [15]. In other words, assume that each n-gram (n) of the encoded record is represented by k bits and the n-grams are almost uniformly distributed over l. For two similar BFs, it is expected (with a high probability) that among m bits, at least $m - k$ bits (considering the Manhattan distance in Equation 3) are identical. This insight requires further investigation and can be explored in future work.

Table 4: Influence of the CRP over the linkage quality.

Dataset	Precision	Recall	F1
mvr	27.5% ±38.1	18.2% ±33.8	23.8% ±35.2
ncvr	11.3% ±14.4	51.2% ±13.5	23.7% ±9.6
dblp_acm	0.6% ±4.0	28.6% ±8.5	18.8% ±4.9
census	15.0% ±22.3	1.7% ±1.8	3.7% ±4.0

The results presented in Table 4 indicate that the CRP parametrization produced (statistically relevant) gains for every tested metric and dataset. However, it is important to notice the limited gain for the census dataset. We believe that this result can be explained by the limitations of the MHT workflow (e.g., the simplified data generation process, which provides only exact matching examples to the classifier) and the complexity of the dataset. In the following research questions, we discuss the limitations of the MHT workflow considering the CRP parametrization presented in Table 3.

6.2.2 Is DLC able to improve the PPRL quality results compared to the baseline and the competitor? To answer this research question, we compare the results of our approach against a baseline and a competitor. As a baseline, we consider the threshold-based classifier. As a competitor, we employ AT-UC [11]. Figure 5 exhibits our results.

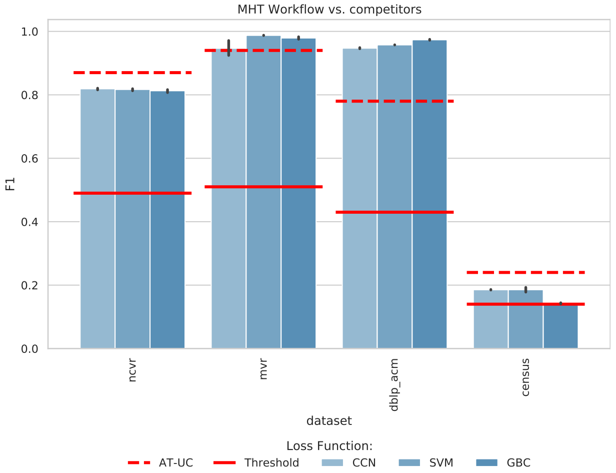


Figure 5: Quality results

The most used classification technique in a PPRL context is the threshold-based classifier. To evaluate this classifier, we tested similarity values from 0.6 to 1.0 with an increment of 0.5 and reported the mean value for the three best results. Comparing DLC against

[5] https://github.com/nobregat/privacy-preserving-learning-encoded-patterns/

the threshold-based classifier, it is possible to recognize that our approach overcomes the mean quality of the baseline.

Comparing DLC and AT-UC, the results show that learning from patterns in the encoded data can substantially improve linkage quality, particularly for the *mvr* and *dblp-acm* datasets. However, the weaker performance observed in the *ncvr* and *census* datasets highlights a limitation of the MHT workflow: the training data generation phase is simplistic and mainly produces exact matches. Consequently, classifiers are trained with limited and less diverse examples, which restricts their ability to capture complex matching patterns — an issue especially evident in the *census* dataset.

Despite this limitation, CRP-based classifiers (e.g., CCN, GBC) consistently matched or outperformed AT-UC. These findings suggest that enhancing training data generation — through data augmentation or privacy-preserving synthesis — represents a promising direction for future work.

6.2.3 *What is the cost in linkage quality incurred during the Privacy-Preserving Training step ($\epsilon > 0$) of the workflow?*

In order to evaluate the comprise between privacy and linkage quality, we executed the MHT workflow for the same input considering different privacy budgets (ϵ) for the MVR dataset. Table 5 shows the impact of the privacy budget over the linkage quality metrics.

Table 5: Linkage quality vs. Privacy budget.

ϵ	Precison	Recall	F1
1.0	+0.61%	-17.47%	-9.81%
5.0	+0.61%	-11.42%	-6.09%
10.0	+0.61%	-2.86%	-1.26%

The results reported in Table 5 demonstrate a clear trade-off between privacy and linkage quality, as tighter privacy budgets degrade model performance. Moreover, for $\epsilon = 1$, a privacy budget capable of maintaining the privacy of the training data in a real-world application, F1 was reduced by almost 10%. In other words, the privacy budget makes it harder for the classifier to extract and identify the patterns due to the noise added to the gradient of the optimizer.

This result was expected and the PPRL parties should be aware of the linkage quality of the privacy budget. Moreover, the PPRL parties must know that the privacy budget makes the classifier lose true match examples.

7 Conclusion and Future Work

This work introduced a methodology that incorporates CRP into a PPRL workflow to represent Bloom filter pairs and train classifiers directly on encoded bit patterns. By moving beyond predefined similarity metrics, DLC enables both deep learning and classical machine learning models to achieve higher linkage quality than threshold-based baselines. Experiments on real-world datasets confirmed the effectiveness of our approach, while also exposing limitations of the current training data generation process, which remains overly simplistic and may introduce bias. Therefore, the proposed workflow reinforces the privacy-aware data collaboration.

For future work, we plan to extend the workflow with techniques such as transfer learning, federated learning, and privacy-preserving data generation, aiming to improve scalability, fairness, and robustness.

8 Acknowledgement

The Article Processing Charge for the publication of this research was funded by the Coordination for the Improvement of Higher Education Personnel (CAPES) (ROR identifier: 00x0ma614) and Post-Graduate Program in Information Technology (PPGTI) of the Federal Institute of Paraíba (IFPB), grant agreement No. 02/2025. The authors have applied a Creative Commons CC BY license for open access.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 308–318.

[2] Tiago Brasileiro Araújo, Vasilis Efthymiou, and Kostas Stefanidis. 2025. Fairness and explanations in entity resolution: an overview. *IEEE Access*.

[3] Andrei Broder and Michael Mitzenmacher. 2004. Network applications of bloom filters: A survey. *Internet Mathematics*, 1, 4, 485–509.

[4] Peter Christen, Thilina Ranbaduge, and Rainer Schnell. 2020. *Linking Sensitive Data*. Springer International Publishing. http://link.springer.com/10.1007/978-3-030-59706-1.

[5] Victor Christen, Tim Häntschel, Peter Christen, and Erhard Rahm. 2023. Privacy-preserving record linkage using autoencoders. *International Journal of Data Science and Analytics*, 15, 4, 347–357. doi:10.1007/s41060-022-00377-2.

[6] Vassilis Christophides, Vasilis Efthymiou, Themis Palpanas, George Papadakis, and Kostas Stefanidis. 2021. An overview of end-to-end entity resolution for big data. *ACM Comput. Surv.*, 53, 6, 127:1–127:42. doi:10.1145/3418896.

[7] Sam Fletcher and Md Zahidul Islam. 2019. Decision tree classification with differential privacy: a survey. *ACM Computing Surveys (CSUR)*, 52, 4, 1–33.

[8] Michael Loster, Ioannis Koumarelas, and Felix Naumann. 2021. Knowledge transfer for entity resolution with siamese neural networks. *Journal of Data and Information Quality (JDIQ)*, 13, 1, 1–25.

[9] Norbert Marwan and Charles L Webber. 2015. Mathematical and computational foundations of recurrence quantifications. In *Recurrence Quantification Analysis*. Springer, 3–43.

[10] Thiago Nóbrega, Carlos Eduardo S Pires, and Dimas Cassimiro Nascimento. 2021. Towards auditable and intelligent privacy-preserving record linkage. In *Anais Estendidos do XXXVI Simpósio Brasileiro de Bancos de Dados*. SBC, 99–105.

[11] Thiago Nóbrega, Carlos Eduardo S. Pires, Dimas Cassimiro Nascimento, and Leandro Balby Marinho. 2023. Towards automatic privacy-preserving record linkage: a transfer learning based classification step. *Data & Knowledge Engineering*, 145, 102180. doi:10.1016/j.datak.2023.102180.

[12] Petri Puustinen, Maria Stratigi, and Kostas Stefanidis. 2025. Stracker: A framework for identifying sentiment changes in customer feedbacks. *Inf. Syst.*, 128, 102491. doi:10.1016/J.IS.2024.102491.

[13] Thilina Ranbaduge, Dinusha Vatsalan, and Ming Ding. 2024. Privacy-preserving deep learning based record linkage. *IEEE Transactions on Knowledge and Data Engineering*, 36, 11, 6839–6850. doi:10.1109/TKDE.2023.3342757.

[14] Alex Rawson, Ewan Duncan, and Conor Jones. 2013. The truth about customer experience. *Harvard business review*, 91, 9, 90–98.

[15] Rainer Schnell. 2016. Privacy-preserving record linkage. In (Feb. 2016), 201–225. doi:10.1002/9781119072454.ch9.

[16] Maria Stratigi, Haridimos Kondylakis, and Kostas Stefanidis. 2020. Multidimensional group recommendations in the health domain. *Algorithms*, 13, 3, 54. doi:10.3390/A13030054.

[17] Dinusha Vatsalan and Peter Christen. 2016. Multi-Party Privacy-Preserving Record Linkage using Bloom Filters, (Dec. 2016). http://arxiv.org/abs/1612.08835.

[18] Dinusha Vatsalan, Peter Christen, and Vassilios S. Verykios. 2013. A taxonomy of privacy-preserving record linkage techniques. *Information Systems*, 38, 6, 946–969. ISBN: 0306-4379. doi:10.1016/j.is.2012.11.005.

[19] Anushka Vidanage, Thilina Ranbaduge, Peter Christen, and Rainer Schnell. 2019. Efficient Pattern Mining based Cryptanalysis for Privacy-Preserving Record Linkage. *Proceedings - International Conference on Data Engineering*, 1698–1701. doi:10.1109/ICDE.2019.00176.

[20] Sumayya Ziyad, Peter Christen, Anushka Vidanage, Charini Nanayakkara, and Rainer Schnell. 2025. A reference set-based encoding method for high-quality and privacy-preserving record linkage. *Information Systems*, 133, 102569. doi:10.1016/j.is.2025.102569.