Linking Patient Records at Scale with a Hybrid Approach Combining Contrastive Learning and Deterministic Rules

Cheng Cao
Truveta Inc.
Bellevue, Washington
chengc@truveta.com

Jay Pillai
Truveta Inc.
Bellevue, Washington
jayadevp@truveta.com

Sina Ghadermarzi
Truveta Inc.
Bellevue, Washington
p-sinag@truveta.com

Sara Daraei
Truveta Inc.
Bellevue, Washington
sarad@truveta.com

Abstract

Linking patient records across disparate healthcare systems is essential to create comprehensive views of patient health, yet this task is complicated by inconsistent identifiers, data quality issues, and privacy constraints. Although traditional deterministic and probabilistic methods have been widely used for record linkage, their performance is often limited in the presence of noisy or incomplete personally identifiable information (PII), and privacy-preserving variants commonly restrict matching to exact token equality. This work presents a hybrid record linkage approach, which integrates a deep embedding model with deterministic rules to leverage both the flexibility and noise-robustness of soft embeddings and reliably and predictable baseline performance from deterministic rules. Using a large-scale real-world dataset, a BERT-based embedding model is fine-tuned using a siamese network with contrastive loss to encode PII fields as numeric vectors. This system is implemented and evaluated on a commercial database consisting of 250 million PII records, showing the successful use of the system in a real-world healthcare setting.

1 Introduction

As healthcare delivery becomes distributed across a multitude of systems and providers, the need to integrate data from disparate sources has become critical. Many health systems in the United States independently collect and store clinical data about patients, often using locally defined identifiers. As patients interact with multiple healthcare institutions, their information becomes fragmented across these silos, resulting in incomplete views of health records and posing challenges for data-driven healthcare delivery and research [18].

Integrating electronic health records (EHRs) across systems can provide a more comprehensive understanding of individual health and improve clinical decision-making, research, and population health initiatives. Consequently, numerous initiatives have emerged to link health records across systems while preserving data quality and patient privacy [1, 2, 4, 11, 12, 14].

Record linkage (RL), also known as entity resolution, seeks to identify and merge records that refer to the same entity. For matching patients, this process typically relies on personally identifiable information (PII), such as names, birth dates, and ZIP codes. However, PII is both privacy-sensitive

Preprint.

and prone to quality issues, including typographical errors and inconsistent or incomplete entries, which complicates direct linkage [8]. As a result recently there has been an intensified interest in privacy-preserving record linkage (PPRL) methods that aim to match records while keeping PII confidential [7]. Common PPRL techniques involve one-way cryptographic hashing of selected PII fields to produce matchable tokens, as employed in networks such as PCORnet [12]. While effective in preserving privacy, these techniques are brittle; exact token matching does not accommodate minor input discrepancies, resulting in diminished recall.

Traditional RL methods are broadly classified as *deterministic*, based on exact matching rules, and *probabilistic*, which use similarity scores to estimate the likelihood of a match [9, 19]. Hybrid methods have also been developed to combine the strengths of both approaches [15]. Recent works often use bloom filters [5, 7, 10, 13, 17], creating a better balance between privacy protection and linkage accuracy, yet challenges persist when applied to noisy real-world data.

Parallel to these developments, machine learning (ML) techniques have been applied to RL tasks, including early work using neural networks [5, 16, 20]. Most ML-based RL models to date, however, rely on relatively simple architectures and small-scale datasets, limiting their practical effectiveness. Transformer-based architectures such as BERT [6] have shown strong performance across various domains by capturing complex relationships in input data through attention mechanisms. Despite this, their application in record linkage remains limited, likely due to the scarcity of large datasets in this domain.

The present study uses a large-scale real-world patient dataset to fine-tune a BERT-based Siamese network [3] using contrastive loss. Patient records are encoded into continuous vector representations (embedding) in which the similarity of the records is reflected by geometric proximity. This embedding-based approach improves robustness to common PII variations—such as misspellings, inconsistent formatting, name changes, and missing values. To harness the strengths of both learned embeddings and deterministic rules, the proposed framework integrates BERT-derived representations with rule-based linkage logic in a hybrid system. This combination achieves strong precision and recall, providing both resilience to data noise and reliability in well-structured matching scenarios.

2 Methods

2.1 Problem Formulation

Let $D = \{r_1, r_2, \ldots, r_n\}$ denote a dataset of patient records (Section 2.2.1 describes the fields that comprise a patient record). The task of record linkage is to identify all pairs (r_i, r_j) such that r_i and r_j refer to the same individual. Formally, this represents a binary classification problem over $n(n-1)/2$ possible record pairs. Let M denote the set of all true match pairs, and $\hat{M}$ the set of predicted matches. The quality of linkage is evaluated using the following standard metrics:

- Precision: $\frac{|M \cap \hat{M}|}{|\hat{M}|}$
- Recall: $\frac{|M \cap \hat{M}|}{|M|}$
- F1 Score: harmonic mean of precision and recall

2.1.1 Blocking

Due to the computational infeasibility of exhaustively comparing all possible record pairs during linkage, a commonly used strategy of blocking is employed. This strategy partitions the dataset into disjoint subsets $B_1, B_2, \ldots, B_k$, referred to as *blocks*, based on the value of a blocking token (a specific concatenation of PII fields). The underlying assumption is that when the value of blocking token is different between two records, the chance of real match is very small. Therefore, the computational and statistical benefits of this exclusion outweigh the potential loss of recall due to missed cross-block matches. Consequently, cross-block comparisons—mostly clearly non-matches—are excluded from them model training, and matching process. This strategy significantly reduces the number of candidate pairs while breaking the problem into smaller, independent subproblems.

Multiple blocking tokens, constructed from various combinations of PII fields, were evaluated, :

- T1: first_name[0] + last_name + DOB + gender

- T2: first_name + last_name + DOB + zip_code
- T3: first_name + last_name + gender + DOB
- T4: first_name + last_name + gender

T1 uses DOB, gender, last name, and only the first letter of the first name—yielding high recall and robustness to noise in the name field. While its precision is moderate, it is sufficient for blocking, where recall is prioritized. T3 includes the full first name, improving precision but reducing recall due to added noise. T2 adds zip code, which is unstable over time, further lowering recall. T4 relies solely on full name and gender, lacks DOB, and performs worst due to low discriminative power.

Empirical analysis (Table 1), backed the reasoning above, and choice of T1 as the candidate that achieved the most favorable trade-off between recall and precision.

2.1.2 Unbiased Estimation of Precision and Recall

Since our embedding-based linkage method operates within blocks, and the evaluation dataset comprises only a subset of these blocks, the observed linkage performance may be overestimated. This is because the likelihood of missing true cross-block linkages—those that are ignored due to the blocking strategy—increases as the number of blocks grows. In this section, we present a method to infer an unbiased estimate of the linkage performance across the entire dataset, accounting for all blocks. *Precision* indicates the proportion of assigned fuzzy IDs that correspond to true matches, as verified by available ground truth such as SSNs. It can be estimated without bias from all evaluated record pairs in the evaluation data that involve distinct entities (i.e., excluding self-to-self comparisons), as it depends only on observed pairs within blocks and does not require estimating cross-block matches:

$$\text{Precision} = \frac{|sameSSN \cap sameID|}{|sameID|}$$

Recall, is defined as the proportion of all true matches—regardless of their blocking status—that are successfully linked by the system. Specifically, this includes both the detected matches within blocks and the undetected cross-block matches. The full recall is defined as:

$$\begin{aligned}\text{Recall} &= \frac{|sameID \cap sameSSN|}{|sameSSN|} \\ &= \frac{|sameID \cap sameSSN|}{|sameSSN_{in}| + |sameSSN_{out}|}\end{aligned}$$

Where:

- $sameSSN_{in}$: pairs with the same SSN that reside within the same block.
- $sameSSN_{out}$: pairs with the same SSN that are located in different blocks.

Due to the quadratic growth of cross-block same-SSN pairs and the limited sampling of blocks during evaluation, direct recall estimation would be biased. To correct for this, the recall is decomposed into two components:

$$\text{Recall} = \left(\frac{|sameID_{in} \cap sameSSN_{in}|}{|sameSSN_{in}|}\right) \times \left(\frac{|sameSSN_{in}|}{|sameSSN|}\right)$$

The first term represents within-block recall and can be directly estimated from the evaluation data without bias. The second term reflects the blocking recall, a fixed proportion measurable from the full dataset. This decomposition enables unbiased recall estimation without extrapolation from incomplete block coverage.

2.2 Dataset

In this work we use a commercial database that is a combination of records from over 30 healthcare systems across the United States and contain records from nearly a fifth of the US population. The dataset used here is a 20% random sample of this database and thus caputres the diversity of patient information.

2.2.1 Features

Figure 1 presents the 13 available personally identifiable information (PII) fields and their respective fill rates, both before and after data cleaning. Among these, five standard PII fields—first name, last name, date of birth (DOB), gender, and ZIP code—exhibited the highest completeness and were therefore selected as the primary features for the linkage model.

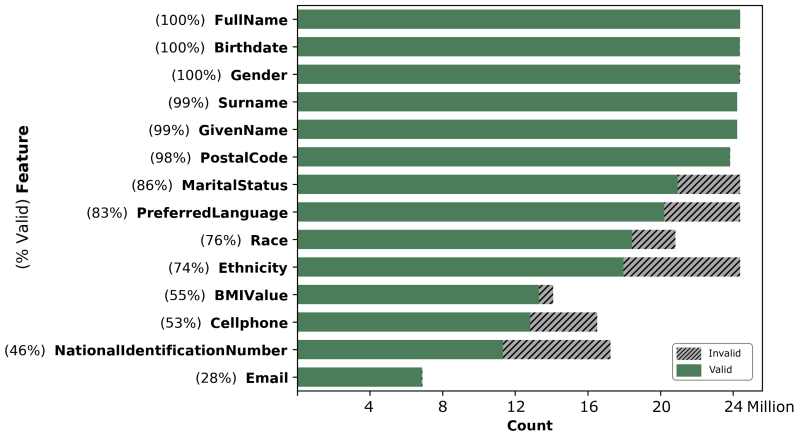

Figure 1: Fill rates for PII features

2.2.2 Ground Truth

Although Social Security Numbers (SSNs) are not universally available and may be affected by data quality issues, valid SSNs—when present—serve as a reliable ground truth anchor. In the absence of manual annotations, valid SSNs are used as a proxy to construct labeled record pairs. To ensure high confidence in the resulting ground truth, after applying a format-check using regular expresions, we manually inspected several hundred SSNs to validate our labeling assumptions. We realized that most of groups of size > 6 are SSNs that are used as a placeholder value (e.g., 111-22-3333) shared by different people. In addition, we observed cases where the same legitimate SSNs were used by multiple members of a family. To filter out these placeholder, or otherwise wrongly shared SSNs, we mark all SSNs that are associated more than one unique DOB and name as invalid. Furthermore, to allow for small variations in DOB and typos in names, we used hamming-distance-based clustering. This process provided a large source of ground-truth labels (1 million records).

2.2.3 Train-Test Split

We assign 20% records to the test set, 10% validation set, and 70% to the training set. To prevent data leakage, the partitioning is performed at the block level, such that all records sharing a blocking token value are assigned to the same partition. This ensures that records in the test set remain disjoint and dissimilar from those in the training set, preserving the integrity of model evaluation.

2.3 Embedding Model

A contrastive loss function is employed to train the embedding model:

- Positive pairs (same patient): L2 distance is minimized
- Negative pairs (different patients): L2 distance is maximized (subject to a margin cap)

Training is conducted using a Siamese network architecture, with input pairs sampled within blocks to avoid information leakage. Redundant pairs arising from symmetric permutations are removed, as pair order does not influence the computed distance.

Each patient record is embedded by converting five key PII attributes—first name, last name, date of birth (DOB), gender, and ZIP code—into JSON format, and then passing this structured input into an uncased PubMedBERT model. The model is configured with 24 attention heads, a maximum

token length of 1,024, and 16-bit precision. Fine-tuning is performed using a contrastive learning strategy that reduces the Euclidean distance between embeddings for positive pairs (records of the same individual), while increasing the distance for negative pairs (records from different individuals).

The following equation defines the contrastive loss function used:

$$\mathcal{L} = y \cdot \|f(x_1) - f(x_2)\|_2^2 + (1-y) \cdot \max\left(0, m - \|f(x_1) - f(x_2)\|_2^2\right) \tag{1}$$

Where:

- x_1, x_2: input pair to the Siamese network.
- $f(\cdot)$: embedding function (i.e., network output).
- $\|f(x_1) - f(x_2)\|_2$: Euclidean distance between embeddings of x_1 and x_2.
- $y \in \{0, 1\}$: label indicating whether the inputs belong to the same class ($y = 1$) or not ($y = 0$).
- m: margin parameter enforcing a minimum separation between dissimilar pairs.
- $\mathcal{L}$: computed contrastive loss.

2.3.1 Training

We dedicate 10% of the data to validation for tuning hyper-parameters and we tune them for best AUC on the validation set, with early stopping, where we use the checkpoint from the last epoch where the AUC on the validation set increases (epoch 7).

For contrastive training, we sample positive and negative pairs only within blocks, and only within pairs that don't have identical PII. Each epoch consist of all possible within-block non-identical pairs, and each batch contain 32 pairs.

2.4 Hybrid Linkage Algorithm

Following the embedding of input PII fields, the next step involves clustering the embeddings and assigning a shared identifier to all records within the same cluster. To further enhance linkage accuracy, additional post-hoc correction steps involving high-confidence deterministic rules are introduced, overriding the potential linkage conflicts between existing ids such as valid social security numbers and embedding-based matchings. These steps are implemented within a hybrid framework, as detailed below.

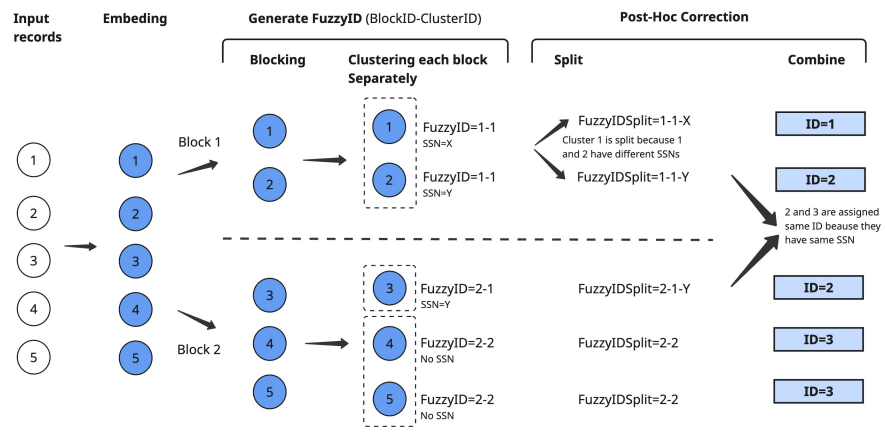

Figure 2: Schematic Illustration of the Hybrid Linkage Algorithm

2.4.1 Generate Fuzzy ID (Blocking and Clustering Step)

The generation of fuzzy IDs begins by dividing records into blocks based on a selected blocking token. Within each block, records are clustered based on their embedding similarity, using a defined Euclidean distance threshold. Blocking serves to eliminate low-probability matches early, while also enabling independent and parallel processing within each block.

After clustering, each element in a block is assigned a temporary fuzzy ID, derived by concatenating the block ID (i.e., the blocking token value) with the cluster number. These fuzzy IDs are subsequently refined using overriding rules in the split and combine stages. Algorithm 1 outlines this process, which takes as input a dataset D with blocking tokens, embeddings, and a distance threshold.

Clustering is performed by constructing a neighborhood graph where nodes (records) are connected if they lie within the distance threshold. The graph is then partitioned into its largest fully connected subgraphs (cliques), which serve as clusters. This ensures that all pairwise distances within a cluster remain within the defined threshold, thereby avoiding linkage propagation artifacts and ensuring high cluster precision.

Algorithm 1: Generate Fuzzy ID

```
// Inputs:
// D: dataset with embeddings and blocking-token T
// T: column containing blocking-token
// thr: embedding distance threshold
Function genFuzzyID(D, T, thr):
    // for each block B with blocking-token value b
    foreach b, B in D.groupBy(T) do
        // create neighborhood graph for the records in B
        G ← neighborhoodGraph(B, thr)
        // divide G into fully-connected subsets (clusters)
        C ← clusters(G)
        foreach cluster c in C do
            /* in each cluster all elements are assigned the same Fuzzy ID: blockID-clusterID */
            setFuzzyID(c, concat(b, c.clusterID))
```

2.4.2 Post-Hoc Corrections Using Deterministic Rules

Once initial fuzzy IDs are assigned, refinement is performed using two types of deterministic rule-based operations:

- Splitting clusters that contain conflicting values for a high-confidence field (e.g., valid SSNs)
- Merging clusters that share a high-confidence field and meet consistency criteria (e.g., same SSN, DOB, and first name)

In the *split* phase, records with different values of a specified *split key* (such as SSN) are required to reside in separate clusters. The presence of differing values for such a key is treated as definitive evidence that the records belong to different individuals. Algorithm 2 describes this procedure, which updates fuzzy IDs based on the values of the split key.

The *combine* phase follows, using a *combine key* (e.g., SSN) for merging. If two records share the same value for a combine key, they are merged into the same cluster. This operation assumes the key provides a sufficient condition for linkage. Algorithm 3 summarizes the process.

Figure 2 illustrates the full hybrid linkage algorithm with an example.

3 Results

3.1 Analyzing Embeddings

To evaluate the effectiveness of embedding distances in distinguishing between matching and non-matching record pairs, the distribution of Euclidean distances is analyzed, as shown in Figure 3. The

Algorithm 2: Postprocess-Split

```
// Inputs:
// D: dataset containing FuzzyID and split key columns
// s: split key
Function split(D, s):
    // for each group G of records with FuzzyID f:
    foreach f, G in D.groupBy(FuzzyID) do
        /* each member either has split key (with) or doesn't (without)*/
        with ← G[G.s ≠ null]
        wihtout ← G[G.s = null]
        /* for members with split key, FuzzyIDSplit will be their FuzzyID attached to their split key*/
        with.FuzyyIDSplit ← concat(f, with.s)
        /* for members without split key, FuzzyIDSplit will be taken from the nearest in the group that has
           split key. */
        if |with| > 0 then
            without.FuzzyIDSplit ← without.nearestIn(with).FuzzyIDSplit
        /* and if there is no member in the group with split key, it will be simply their FuzzyID. */
        else
            without.FuzzyIDSplit ← f
```

Algorithm 3: Postprocess-Combine

```
// Inputs:
// D: dataset containing FuzzyIDSplit and combine key columns
// c: combine key
Function combine(D, c):
    /* groups of records with same value of combine key c will get the same ID (FuzzyIDSplit with highest
       number in the group) */
    foreach k, G in D.groupBy(c) do
        if |G| > 0 then
            G.ID ← findMajority(G.FuzzyIDSplit)
```

histogram reveals a clear separation between positive and negative pairs, indicating that the learned embedding space successfully encodes semantic similarity. It is important to note that the evaluated pairs belong to the same block and thus already share substantial PII-based similarity.

Figure 4 shows an alternative illustration of the discriminative power of embedding distance by plotting the receiver operating characteristic (ROC) curve. The area under the curve reflects strong predictive utility in differentiating matched from unmatched record pairs based on learned distances.

3.2 Matching Performance

Reliable evaluation of matching performance typically requires ground-truth annotations, which are often manually curated. Given the scale of the dataset used in this work, manual annotation was infeasible. As an alternative, valid SSNs are employed as a proxy for ground truth during evaluation.

Applying the full hybrid framework—using SSN as both the split and combine key—results in perfect performance, with precision and recall of 100%. However, to isolate the contribution of the embedding-based clustering component, Table 1 reports performance metrics based on fuzzy IDs generated prior to the split and combine stages.

The results demonstrate that the fuzzy ID approach alone significantly outperforms traditional baselines. Precision, recall, and F1 score are reported for various configurations, including exact string match and token-based rules T1–T4 as defined in Section 2.1.1.

Prec: Precision, *Rec:* Recall. The column labeled "Exact" reflects exact string match of PII features. The column "T1–T4" represents matching that requires all token rules T1 through T4 to match.

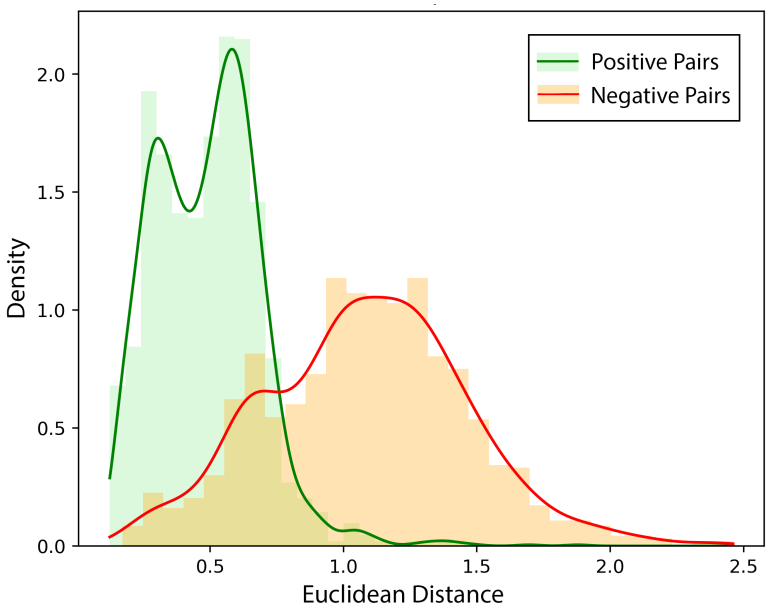

Figure 3: Distribution of embedding distance in positive and negative pairs

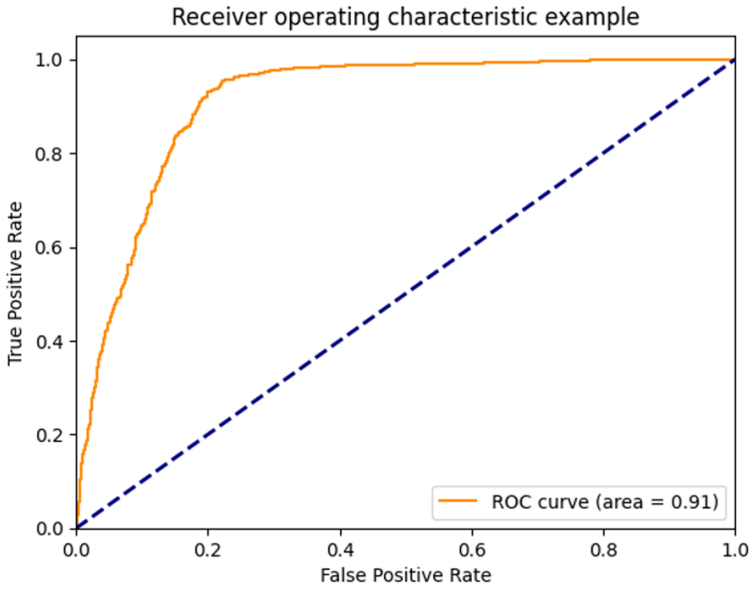

Figure 4: ROC of the embedding distance as a predictor of label

Table 1: Matching Performance for Fuzzy ID (pre-Split/Combine) and Baselines

	FuzzyID	Exact	T1	T2	T3	T4	T1-T4
Prec.	**0.94**	0.98	0.81	0.97	0.95	0.004	0.97
Rec.	**0.93**	0.65	0.95	0.69	0.82	0.85	0.69
F1	**0.935**	0.782	0.874	0.86	0.880	0.008	0.806

4 Discussion

The proposed hybrid framework integrates the adaptability of deep embedding models with the precision and interpretability of deterministic rules. This combination offers a robust solution to challenges associated with data noise, privacy requirements, and computational efficiency in large-scale healthcare record linkage. The learned embeddings can be utilized as privacy-preserving identifiers and may also serve as inputs for secure multiparty computation protocols.

5 Conclusion

This study presents a scalable and privacy-aware record linkage framework that leverages contrastive BERT embeddings in conjunction with deterministic rule-based post-processing. The proposed method achieves state-of-the-art accuracy on a large-scale, real-world patient dataset, demonstrating strong generalization performance and robustness in noisy, privacy-sensitive environments.

References

[1] Majid Afshar, Madeline Oguss, Thomas A Callaci, Timothy Gruenloh, Preeti Gupta, Claire Sun, Askar Safipour Afshar, Joseph Cavanaugh, Matthew M Churpek, Edwin Nyakoe-Nyasani, Huong Nguyen-Hilfiger, Ryan Westergaard, Elizabeth Salisbury-Afshar, Megan Gussick, Brian Patterson, Claire Manneh, Jomol Mathew, and Anoop Mayampurath. Creation of a data commons for substance misuse related health research through privacy-preserving patient record linkage between hospitals and state agencies. *JAMIA Open*, 6(4):ooad092, December 2023.

[2] Jiang Bian, Alexander Loiacono, Andrei Sura, Tonatiuh Mendoza Viramontes, Gloria Lipori, Yi Guo, Elizabeth Shenkman, and William Hogan. Implementing a hash-based privacy-preserving record linkage tool in the OneFlorida clinical research network. *JAMIA Open*, 2(4):562–569, December 2019.

[3] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *Advances in Neural Information Processing Systems*, volume 6, pages 737–744. Morgan Kaufmann, 1993.

[4] Victor M Castro, Vivian Gainer, Nich Wattanasin, Barbara Benoit, Andrew Cagan, Bhaswati Ghosh, Sergey Goryachev, Reeta Metta, Heekyong Park, David Wang, Michael Mendis, Martin Rees, Christopher Herrick, and Shawn N Murphy. The Mass General Brigham Biobank Portal: an i2b2-based data repository linking disparate and high-dimensional patient data to support multimodal analytics. *Journal of the American Medical Informatics Association*, 29(4):643–651, April 2022.

[5] Victor Christen, Tim Häntschel, Peter Christen, and Erhard Rahm. Privacy-preserving record linkage using autoencoders. *International Journal of Data Science and Analytics*, 15(4):347–357, May 2023.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186. Association for Computational Linguistics, 2019. arXiv:1810.04805.

[7] Aris Gkoulalas-Divanis, Dinusha Vatsalan, Dimitrios Karapiperis, and Murat Kantarcioglu. Modern Privacy-Preserving Record Linkage Techniques: An Overview. *IEEE Transactions on Information Forensics and Security*, 16:4966–4987, 2021. Conference Name: IEEE Transactions on Information Forensics and Security.

[8] Shaun J Grannis, Huiping Xu, Joshua R Vest, Suranga Kasthurirathne, Na Bo, Ben Moscovitch, Rita Torkzadeh, and Josh Rising. Evaluating the effect of data standardization and validation on patient matching accuracy. *Journal of the American Medical Informatics Association*, 26(5):447–456, May 2019.

[9] Erel Joffe, Michael J Byrne, Phillip Reeder, Jorge R Herskovic, Craig W Johnson, Allison B McCoy, Dean F Sittig, and Elmer V Bernstam. A benchmark comparison of deterministic and probabilistic methods for defining manual review datasets in duplicate records reconciliation. *Journal of the American Medical Informatics Association*, 21(1):97–104, January 2014.

[10] Dimitrios Karapiperis, Aris Gkoulalas-Divanis, and Vassilios S. Verykios. FEDERAL: A Framework for Distance-Aware Privacy-Preserving Record Linkage. *IEEE Transactions on Knowledge and Data Engineering*, 30(2):292–304, February 2018.

[11] Abel N Kho, John P Cashy, Kathryn L Jackson, Adam R Pah, Satyender Goel, Jörn Boehnke, John Eric Humphries, Scott Duke Kominers, Bala N Hota, Shannon A Sims, Bradley A Malin, Dustin D French, Theresa L Walunas, David O Meltzer, Erin O Kaleba, Roderick C Jones, and William L Galanter. Design and implementation of a privacy preserving electronic health record linkage tool in Chicago. *Journal of the American Medical Informatics Association*, 22(5):1072–1080, September 2015.

[12] Daniel Kiernan, Thomas Carton, Sengwee Toh, Jasmin Phua, Maryan Zirkle, Darcy Louzao, Kevin Haynes, Mark Weiner, Francisco Angulo, Charles Bailey, Jiang Bian, Daniel Fort, Shaun Grannis, Ashok Kumar Krishnamurthy, Vinit Nair, Pedro Rivera, Jonathan Silverstein, and Keith Marsolo. Establishing a framework for privacy-preserving record linkage among electronic health record and administrative claims databases within PCORnet®, the National Patient-Centered Clinical Research Network. *BMC Research Notes*, 15(1):337, October 2022.

[13] Ibrahim Lazrig, Toan C. Ong, Indrajit Ray, Indrakshi Ray, Xiaoqian Jiang, and Jaideep Vaidya. Privacy Preserving Probabilistic Record Linkage Without Trusted Third Party. In *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, pages 1–10, Belfast, August 2018. IEEE.

[14] Keith Marsolo, Daniel Kiernan, Sengwee Toh, Jasmin Phua, Darcy Louzao, Kevin Haynes, Mark Weiner, Francisco Angulo, Charles Bailey, Jiang Bian, Daniel Fort, Shaun Grannis, Ashok Kumar Krishnamurthy, Vinit Nair, Pedro Rivera, Jonathan Silverstein, Maryan Zirkle, and Thomas Carton. Assessing the impact of privacy-preserving record linkage on record overlap and patient demographic and clinical characteristics in PCORnet®, the National Patient-Centered Clinical Research Network. *Journal of the American Medical Informatics Association*, 30(3):447–455, March 2023.

[15] Toan C Ong, Lindsey M Duca, Michael G Kahn, and Tessa L Crume. A hybrid approach to record linkage using a combination of deterministic and probabilistic methodology. *Journal of the American Medical Informatics Association*, 27(4):505–513, April 2020.

[16] Thilina Ranbaduge, Dinusha Vatsalan, and Ming Ding. Privacy-Preserving Deep Learning Based Record Linkage. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6839–6850, November 2024.

[17] Kurt Schmidlin, Kerri M. Clough-Gorr, Adrian Spoerri, and for the SNC study group. Privacy Preserving Probabilistic Record Linkage (P3RL): a novel method for linking existing health-related data and maintaining participant confidentiality. *BMC Medical Research Methodology*, 15(1):46, December 2015.

[18] Kanan Shah, Debra Patt, and Samyukta Mullangi. Use of Tokens to Unlock Greater Data Sharing in Health Care. *JAMA*, 330(24):2333, December 2023.

[19] Miranda Tromp, Anita C. Ravelli, Gouke J. Bonsel, Arie Hasman, and Johannes B. Reitsma. Results from simulated data sets: probabilistic record linkage outperforms deterministic record linkage. *Journal of Clinical Epidemiology*, 64(5):565–572, May 2011.

[20] D. Randall Wilson. Beyond Probabilistic Record Linkage: Using Neural Networks and Complex Features to Improve Genealogical Record Linkage. In *The 2011 International Joint Conference on Neural Networks*, pages 9–14, July 2011. ISSN: 2161-4407.