UNIVERSITY OF MANNHEIM
School of Business Informatics and Mathematics

Chair of Practical Computer Science IV:
Dependable Systems Engineering

Bachelor's Thesis

A Study on the Characteristics of a Universal, Unsupervised Machine Learning Graph Matching Attack to Increase Linkage Success in Privacy-Preserving Record Linkage

as part of the degree program Bachelor of Science Business Informatics submitted by

Kilian Hüllen
Matriculation number 1749251

on July 16, 2024.

Supervisor: Prof. Dr. Frederik Armknecht
M. Sc. Jochen Schäfer

Abstract

Privacy-preserving record linkage (PPRL) aims to identify entries across different datasets that refer to the same real-world entity without disclosing personally identifiable information (PII) to external parties. This can be achieved by encoding the PII in a similarity-preserving way, and subsequently linking records together based on the similarity of their encoded representations. The fact that similarities between records must be preserved though, enables the use of graph matching attacks (GMAs) to corrupt the data privacy and re-identify encoded records.

Recently, a new GMA approach has been proposed by Schäfer et al. [SA24] that is based on unsupervised machine learning and outperforms previous GMAs significantly. This thesis aims to examine the properties, possible enhancements and use cases of this newly introduced GMA in depth. For this, four experiments are conducted to examine the following aspects: the influence of using a known ground truth on the linkage success, a comparison of different bipartite graph matching techniques used within the GMA with regards to maximum linkage success and minimum false positives rate, the runtime behaviour for increasingly larger datasets, and the influence of erroneous data on the linkage success.

An analysis of the experiments shows that only the knowledge about equal encodings across the datasets does not enhance the linkage success. When adding dummy values to both datasets for a better graph alignment however, the linkage success can be improved significantly. Bipartite graph matching techniques ensuring a full 1-1 mapping between records from both datasets are the most stable and reach the highest linkage success. Their respective false positives rates only decrease linearly though for increasingly larger overlaps between the two datasets. The second step of the GMA consisting of embedding the nodes in the similarity graphs to enable proper alignment and distance measurements, takes around $\frac{2}{3}$ of the whole time of the algorithm. In addition, the unsupervised alignment step can be very time consuming for unfavourable start values. Lastly, as to be expected, high error rates in one of the two datasets lead to a decrease in linkage success and make the results less stable.

Contents

List of Figures

List of Tables

Acronyms

BF Bloom filter

FPR false positives rate

GMA graph matching attack

LSR linkage success rate

LU linkage unit

ML machine learning

MRO minimum required overlap

NN nearest neighbour

ONS Office for National Statistics

PII personally identifiable information

PPRL privacy-preserving record linkage

QID quasi identifier

RL record linkage

TMH tabulation MinHash

TSH two-step hashing

UID unique identifier

1. Introduction

Record linkage (RL) describes the process of identifying, and consequently linking entries across different databases together that refer to the same real world entity, often a human individual [FKP12]. On an individual scale, its goal is to increase knowledge about an individual by gathering information from a variety of data sources and storing them together. On a more general scale though, RL can also provide a first, valuable step for statistical research, customer behaviour prediction or fraud detection by enlarging the data basis for subsequent analyses. Thus, RL has applications in administrative areas like marketing, data warehousing, law enforcement and governmental services, as well as for research purposes in, amongst others, the fields of epidemiology, healthcare and social science research [Gu+03].

However, when sensitive or personally identifiable information (PII) is shared with external subjects, privacy concerns arise, calling for the need of protocols and techniques to ensure data protection and preserving privacy while still being able to identify records referring to the same individual. Privacy-preserving record linkage (PPRL) techniques have been developed to meet these requirements.

In PPRL, the PII of entries in databases is encoded, and only the similarities between encoded PII is used to decide which records most likely refer to the same individual. This allows for matching of the anonymised data without disclosing PII to third parties. However, the inherent need of PPRL to compute similarities between records enables so-called graph matching attacks (GMAs). By creating two similarity graphs, each representing the similarities between records in one dataset respectively, and subsequently aligning the graphs together as accurately as possible, record pairs referring to the same individual can be identified. If an attacker makes use of a publicly available dataset whose PII is not encoded, and aligns its corresponding similarity graph to the graph of a previously encoded, sensitive dataset, they can see the PII of every record pair, thus corrupting the attempts of PPRL to preserve privacy. Therefore, GMAs pose the biggest threat to PPRL these days.

While different GMA approaches have been proposed in recent years, they all make partially unrealistic assumptions by requiring high attacker knowledge. A new proposal by Schäfer et al. [SA24] based on unsupervised machine learning (ML) though has been proven to outperform previous GMAs significantly while at the same time assuming very limited attacker knowledge. This potentially makes GMAs significantly more applicable in real-world scenarios, posing a realistic and immediate threat to established PPRL techniques.

1.1. Contribution

While the new GMA by [SA24], GMA_{ML}, provided promising results, it is still subject to potential improvements, especially for those scenarios where it has not yet performed very well. Additionally, many key properties and behaviours of GMA_{ML} are still unknown, and further study of these will contribute to and open up further advancements in PPRL and GMAs.

For this reason, this thesis examines four properties of and scenarios related to GMA_{ML} with the aim of getting a better understanding of the attack and improving its effectiveness for different situations. However, these examined situations are not exclusively applicable to the malicious use of the algorithm as a GMA, but should also examine the effectiveness of GMA_{ML} for PPRL in its original, non-malicious form. As previously proposed GMAs are only applicable very restrictively due to their many assumptions, and GMA_{ML} proved to be very effective, this work can provide an important contribution to the most recent developments in PPRL and GMA techniques and best practices.

First, the influence of a known ground truth on the RL success is examined, i.e., if the initial knowledge about some records belonging together can increase the number of correctly matched records. As part of this, the possibility of adding a dummy dataset to both datasets to be linked, with the aim of enabling better graph alignment, is also considered and evaluated.

Secondly, four different bipartite graph matching approaches are compared. Once the two similarity graphs have been aligned, it has to be decided which records, represented by nodes in the two similarity graphs, should ultimately be paired together. For this, minimum weight graph matching, matching by solving the stable marriage problem, symmetric as well as nearest neighbour (NN) matching are considered and evaluated based on the number of correctly matched records, and by their false positives rate (FPR), i.e., how many records have been matched incorrectly.

Third, the runtime of GMA_{ML} is examined for increasingly larger datasets to assess how applicable the algorithm is to a variety of real-world scenarios requiring to process different volumes of data.

Lastly, the influence of erroneous data on the linkage success and stability of GMA_{ML} is analysed using modified records with an error rate of 2%, 5%, 10%, 15% and 20% respectively as input for the second dataset in every test run.

The complete code repository can be found on GitHub[1].

1.2. Organization of this Thesis

To examine the properties and situations explained above, the outline of this thesis is as follows: In chapter 2, the theoretical background needed for the conducted experiments is covered. Chapter 3 gives an overview of relevant work on GMAs, on which this thesis is based. Afterwards, chapter 4 contains detailed descriptions of the setups for the four experiments on GMA_{ML}, and in chapter 5, these experiments are analysed and discussed. The thesis concludes with a summary of the achieved work and an outlook for further work in the future in chapter 6.

[1] https://github.com/kilian1322/pprl-gma-ml-ba

2. Background

This section provides a brief overview over RL in general and PPRL in particular, as well as the key concepts of GMAs. Additionally, the techniques and algorithms used in GMA_{ML} are explained to provide the technical background needed to follow the conducted experiments described and analyzed in chapter 4 and 5.

2.1. Linking Database Records

Record linkage (RL) describes the process of identifying records in different datasets that refer to the same real-world entity and was first introduced in 1946 by Dunn [Dun46]. In most cases, RL is carried out on two different databases and the stored entries (records) refer to individuals in a specific role, e.g. medical patients, social network users or citizens. However, RL can also be performed across more than two databases or on entries within one single database.

In its simplest form, RL can be performed by directly comparing unique identifiers (UIDs) such as social security or medical record numbers. Additionally, quasi identifiers (QIDs), e.g., name, date of birth, zip code, gender or occupation, can be used to link records. In contrast to UIDs, these attributes alone cannot generally identify an individual, but combined they have the potential to distinguish uniquely between the different entities stored in datasets [GD+21].

The general process of record linkage consists of several steps that can be seen in Fig. 2.1. First, the database owners O_1 and O_2 individually perform data pre-processing operations on their data to increase data quality. This step usually consists of data cleansing in order to deal with missing values, duplicates and inconsistencies in data types and representations that might occur. A basic form of data cleansing can be performed by O_1 and O_2 independently from each other. Nonetheless, for maximum data quality, it is advantageous for O_1 and O_2 to coordinate this step together. This ensures that the PII used in the actual RL step later on is as homogeneous and consistent across both databases as possible.

An optional second step for a complete and successful linkage process consists of indexing . However, for increased efficiency, especially for large datasets, this step is highly recommended. Indexing aims at blocking or filtering the data to reduce the comparisons needed in the actual linkage step. This can be done by sorting and thereby classifying the records based on one or more of their attributes. Given a certain data quality and thus assuming that all entries are classified correctly, entries in database D_1 with an attribute A_1 and a specific attribute value a_1 only have to be compared to entries having the same attribute value a_1 in D_2. Therefore, a comparison of an entry in D_1 with a_1 does not need to be compared to entries in D_2 with an attribute value of $a_{2,3,\ldots}$ as these values most likely do not refer to the same entity.

Afterwards, the comparison itself can be performed. In this step, all potential matches are compared towards their similarity using one or several similarity functions. To allow for possible errors in the data, a trivial integer, string or object comparison for equality is often not sufficient. Instead, similarities of attributes are calculated using metrics like the Levenshtein distance [Lev+66], or the Jaccard or Dice coefficient [TJ13].

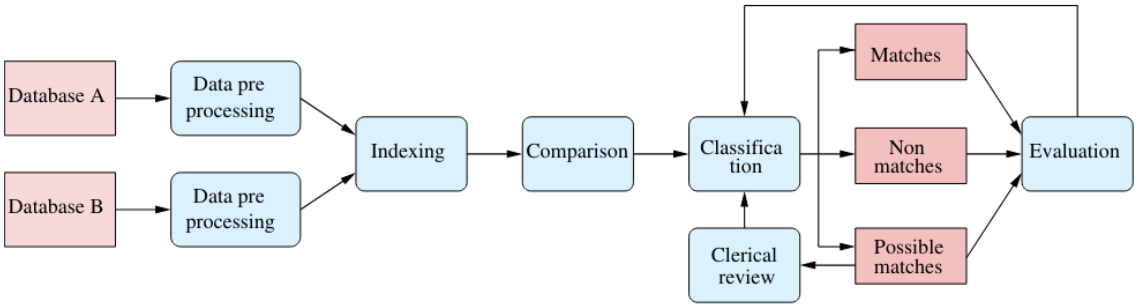

Figure 2.1.: Overview of the General Process of Record Linkage, Source: [VCV13], p. 948

When the similarities between potential matches have been calculated, a decision model can assign matching and non-matching records as well as declare record pairs as possible matches that would need to be investigated further [GB06].

The comparison and classification can either be done by O_1 and O_2 bilaterally, or through an independent third party, a so-called linkage unit (LU). In the latter case, O_1 and O_2 share their data with the LU which then performs the comparison and classification. In the end, the LU returns the tuples of matching records to O_1 and O_2.

2.1.1. Making Record Linkage Privacy-Preserving

Research on RL, also for heterogeneous or inconsistent data, can be dated back to the 1940s. However, a direct comparison of UIDs or QIDs is often not desirable or legal as PII is revealed to untrustworthy or unauthorized parties. This disclosure of PII can occur if, e.g., the second involved data owner does not meet legal standards or if the LU is corrupted Therefore, the need to perform RL while at the same time being able to ensure privacy and sensitive data protection arose. Since the late 1990s, research has been conducted on privacy-preserving record linkage (PPRL) in order to avoid sharing PII with these potentially problematic third parties [GD+21].

In the PPRL process, the PII, i.e., all attributes that can potentially re-identify an individual, are being encoded before the data is shared with other database owners or third parties. In addition to ensuring that the encoded PII cannot be reconstructed, the chosen encoding method must also fulfill the condition that its output data can be compared for similarity in such a way that these similarity values allow direct conclusions about the similarities of the original PII. The three most prevalent encoding techniques in PPRL are Bloom filters (BFs), tabulation MinHash (TMH) and two-step hashing (TSH). They will be discussed in more detail in Section 2.3.

Similarly to the simple RL, in PPRL the records can then be compared using similarity metrics such as the Dice or Jaccard coefficient, and pairs of records can be classified as matches, non-matches or potential matches by a decision model.

2.2. Breaking PPRL: Graph Matching Attacks

Efforts from researchers and malicious actors to corrupt the protection of private data in PPRL have shown conceptual and design weaknesses and strengthened encoding methods and

protocols over the last two decades.

The most prevalent type of attack against PPRL are graph matching attacks (GMAs). These attacks exploit the mere fact that records are being matched based on their similarity to other records, i.e., the inherent need of PPRL to compute pairwise similarities between different records - whether published as plain-text data or encrypted.

A generic GMA approach can be described as follows: Suppose an attacker has gotten access to the records of a private database D_{private} where the PII of each record is encoded. Next, the attacker can use a publicly available database D_{public} as reference. For a successful re-identification using GMAs, the PII in D_{private} and D_{public} must overlap and ideally be the same. More specifically, the UID or QIDs from both datasets should be identical to enable a high re-identification rate. From an attacker's point of view, the PII of D_{private} is encoded. As part of the attack, the records in D_{public} can also be encoded by the attacker to recreate the influence the encoding step has on the similarity relations of D_{private} as accurately as possible.

Almost all organizations, companies and public service providers nowadays store the same QIDs like name, date of birth or zip code for individuals [Cur18; Est17]. Thus, even if the attacker has no knowledge over the PII used to form the set of QIDs of D_{private}, in the worst case they can iterate over the power set of all commonly used and in D_{public} available QIDs, perform one GMA for every combination of QIDs and keep the linkage results of the attack that performed best.

Another property of GMAs that has to be kept in mind is that individuals from the encoded database can only be re-identified if a record referring to that exact individual is also present in the plain-text database D_{public} used by the attacker. While the attacker cannot guarantee that $D_{\text{private}} \subseteq D_{\text{public}}$, i.e., that every individual from the private, encoded database is actually represented in the publicly available database, sources like population registries or commercial data brokers offer extensive sources of data that promise a high number of overlaps.

Given these two databases D_{private} and D_{public}, an attacker first creates two similarity graphs G_{private} and G_{public} independently from one another. This can be done using above mentioned similarity metrics like the Levenshtein distance, or the Dice or Jaccard coefficient. In the similarity graph, every node represents one record, and the weighted edges between nodes relate directly to their respective similarities. Fig. 2.2 visualizes the generation of G_{private} and G_{public}. Although for every tuple of nodes within one graph, a similarity score can be calculated, one can see that some nodes are not directly connected through an edge. Due to efficiency reasons, many GMAs sort out the edges with low weights, leaving only those that suggest that the two connected nodes are similar to each other to at least a certain degree.

Then, in the crucial graph matching step itself, the nodes of the two graphs are compared with each other. Using the underlying idea that within one graph, each node is uniquely characterized by its neighbourhood structure, i.e., its distance to other nodes, one can expect the records from D_{private} and D_{public} referring to the same individual to have very similar neighbourhood structures.

In the final step of record re-identification, the attacker then performs bipartite graph matching [Cro16]. During this process, based on a specified metric, one aims to assign every node in G_{private} to at most one node in G_{public}. These assigned tuples are interpreted as referring to the same individual.

The attacker still does not know any PII from D_{private} as they remain encoded. However, they can now associate the non-encoded data stored in D_{private} with the personal identifiers, e.g., a person's full name, from D_{public}, and thus corrupting and breaking the individuals' privacy.

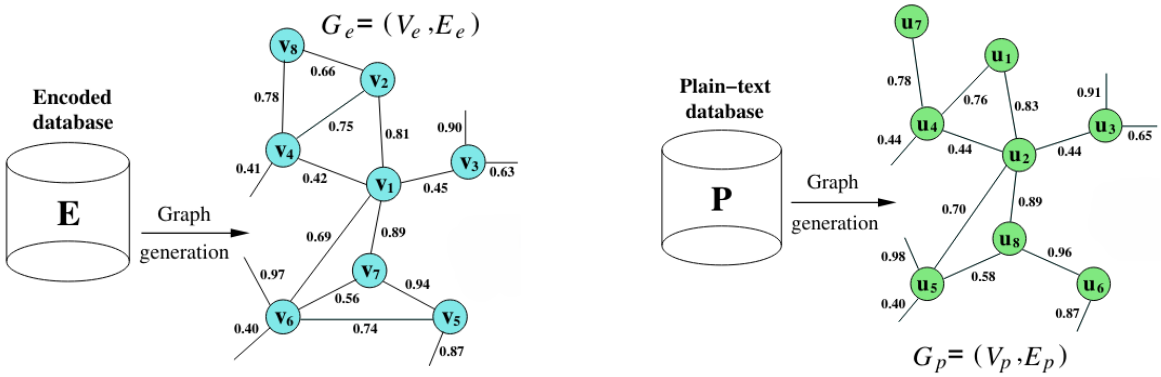

Figure 2.2.: Similarity Graph Creation for Private and Public Database, Source: [Vid+20], p. 1486

The experiments conducted in this thesis are based on one specific GMA implemented by [SA24]. Their approach, as well as other relevant GMA implementations, will be explained in more detail in chapter 3. [SA24] use BFs, TMH and TSH for encoding their test data. Afterwards, they create two similarity graphs and then use the Node2Vec algorithm to map the nodes to vectors in a low-dimensional feature space for robustness [GL16]. In the embedding alignment step, the Wasserstein distance between vectors is used to linearly transform one graph into the space of the other one [GJB19]. Finally, bipartite graph matching is being performed using one of these techniques: minimum weight graph matching, matching through solving the stable marriage problem, symmetric graph matching and NN matching. These techniques and algorithms will be explained in the following subsections before continuing with Related Work and own conducted experiments.

2.3. Encoding Techniques in PPRL

For PPRL, the major difference to RL in its original, simplest sense is that all PII is encoded before sharing it with anyone. However, as already mentioned, in the realm of PPRL one cannot simply use any encoding technique. While the input data must be encrypted irreversibly so that an attacker cannot simply decrypt the PII, it must also hold that the similarity relations between records are roughly preserved. More specifically, the similarity between two encoded records $sim(r_{\text{enc},1}, r_{\text{enc},2})$ must provide an estimation about the similarity between the original data $sim(r_1, r_2)$; thus providing evidence if r_1 and r_2 refer to the same individual.

In research and practice, three encoding techniques have been used and discussed over the last years: Bloom filters, tabulation MinHash, and most recently two-step hashing [SA24]. The following subsections provide an overview of these encoding methods as well as their strengths and weaknesses.

2.3.1. Bloom Filters

BF encoding aims at converting any input into a BF b of length l bits, i.e., a bit vector $b \in \{0,1\}^l$. The relevant parameters are l, as well as the n-gram value q and the number of

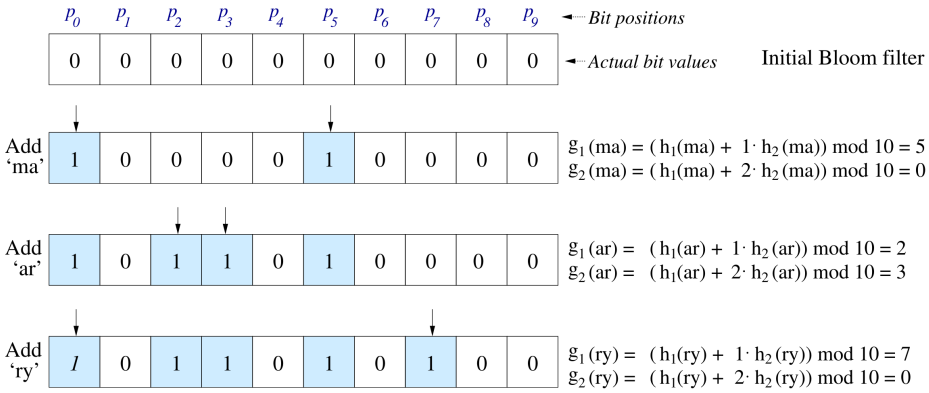

Figure 2.3.: Example of Hashing the Set $S = \{ma, ar, ry\}$ into a Bloom filter for $q = 2$, $k = 2$ and $l = 10$ using Double Hashing for the Input String 'mary', Source: [CRS22], p. 198

used hash functions to map the input values to the BF indices $[0, \ldots, l-1]$, k.

The input value, usually a pure string attribute or the concatenation of several string values, is first converted into a set $S = \{s_1, s_2, \ldots, s_n\}$ of substrings of length q each, also called q-grams. This can be done using the sliding windows approach. For example, the string 'alice' is converted into the list of 2-grams $[al, li, ic, ce]$ for $q = 2$, or the list of 3-grams $[ali, lic, ice]$ for $q = 3$.

Next, the q-grams in S are hashed separately into the BF. The number of independent hash functions used is determined by the BF encoding parameter k. Additionally, the hashing method can be chosen. In the context of PPRL, well-known methods like Double Hashing [DM04b], Triple Hashing [DM04a] or Random Hashing [SB16] are commonly used. Every q-gram is hashed to one position $[0, \ldots, l-1]$ of the BF. At the beginning, every position of the BF, i.e., every bit in the bit vector b, is set to 0. When a character of a q-gram is hashed to the index $i \in \{0, \ldots, l-1\}$, the bit at this position in b is permanently flipped to 1. An example procedure of how a set of three 2-grams is hashed into a BF can be seen in Fig. 2.3.

After having hashed two sets S_1 and S_2 into BFs, one can calculate their similarity using set based similarity functions like the Jaccard or Dice coefficient. Commonly, the Dice coefficient is used as it is insensitive to matching 0-bits in long BFs. For two BFs b_1 and b_2, the Dice coefficient similarity can be calculated as follows:

$$sim_{\mathrm{D}}(b_1, b_2) = \frac{2 \cdot c}{x_1 + x_2}$$

where c is the number of bit positions that have been set to 1 in both BFs, and x_1 and x_2 are the number of 1-bits in b_1 and b_2 respectively [CRS22].

As the n-grams in S are usually directly based on the PII attributes of the records in a dataset D, and as the positions of 1-bits in the respective BF b are directly related to these n-grams, similarity metrics calculations on the encoded data therefore allows direct conclusions about similarities in the original, non-encoded data. However, this makes BF encoding vulnerable to frequency attacks [Chr+18; Vid+19].

2.3.2. Tabulation MinHash

TMH uses the MinHash algorithm [Bro97] for fast set similarity estimations with tabulation-based hashing methods. This class of hashing methods provides efficient look-up tables for calculated hash values [Tho17]. For MinHash, this property becomes highly valuable as a large number of hash values needs to be calculated:

MinHash aims to approximate the Jaccard similarity coefficient $sim_{\mathrm{J}}(s_1, s_2)$ of two sets s_1 and s_2. This estimation is achieved by MinHash because the probability of two sets s_1 and s_2 to generate the same MinHash value equals the Jaccard coefficient of s_1 and s_2 if the MinHash values are computed for a sufficiently large number of hash functions [CRS22]. The Jaccard coefficient therefore is defined as follows:

$$sim_{\mathrm{J}}(s_1, s_2) = \frac{|s_1 \cap s_2|}{|s_1 \cup s_2|} = p[h(s_1) = h(s_2)]$$

for a hash function h.

When using k hash functions $\{h_1, h_2, \ldots, h_{\mathrm{k}}\}$ in succession, the Jaccard coefficient can be estimated through

$$sim_{\mathrm{J_{est}}}(s_1, s_2) = \frac{\sum_{i=1}^{k} [h_{\mathrm{i}}(s_1) = h_{\mathrm{i}}(s_2)]}{k} \approx sim_{\mathrm{J}}(s_1, s_2)$$

where $[h_{\mathrm{i}}(s_1) = h_{\mathrm{i}}(s_2)]$ returns 1 and $[h_{\mathrm{i}}(s_1) \neq h_{\mathrm{i}}(s_2)]$ returns 0.

The MinHash value of each hash function h_{i} is calculated by applying a random permutation π_{i} that represent h_{i} on s_1 and s_2. Then, the first values $\pi_{\mathrm{i}}(s_1)[0]$ and $\pi_{\mathrm{i}}(s_2)[0]$ of this permutation are compared. These first values represent the MinHash values of s_1 and s_2 for a h_{i}. This is done for all k hash function to estimate $sim_{\mathrm{J}}(s_1, s_2)$. MinHash is only an estimation, but intuitively, the greater the similarity between s_1 and s_2, the more MinHash values are equal for the calculated permutations. Thus, for a large number of applied hash functions, the Jaccard coefficient can be estimated well.

TMH is more secure than BF encoding. However, it is significantly more costly in terms of time and space complexity [Smi17].

2.3.3. Two-Step Hashing

TSH aims to provide an encoding mechanism addressing both, the insecurity of BFs and the high demand in space and time complexity of TMH [RCS20]. An example run of TSH can be seen in Fig. 2.4. First, the input strings s_1 and s_2 are split into n-grams Q_1 and Q_2 once again. Afterwards, the n-grams are hashed into k BFs of length l, each time with a different hash function $h_{1,\mathrm{i}}, 1 \leq i \leq k$.

As this BF encoding alone has been proven to be susceptible to dictionary attacks [Mit+17], this is followed by a second hashing step. For this, the column vectors that contain at least one 1-bit value, are hashed to integer values e_{j} using hash functions $h_{2,\mathrm{i}}, 1 \leq i \leq l$. For increased security, and to avoid two columns at different positions in the bit matrix to be hashed to the same integer value, the column index p and a secret salt are added prior to the second hashing. Prior to the encoding step, the two database owners O_1 and O_2 must have agreed on the salt. The two sets of all these integers e_{j}, E_1 and E_2, of the two input strings s_1 and s_2 can then be compared for similarity using the Jaccard coefficient in its original form for set similarities:

$$sim_{\mathrm{J}}(E_1, E_2) = \frac{|E_1 \cap E_2|}{|E_1 \cup E_2|}$$

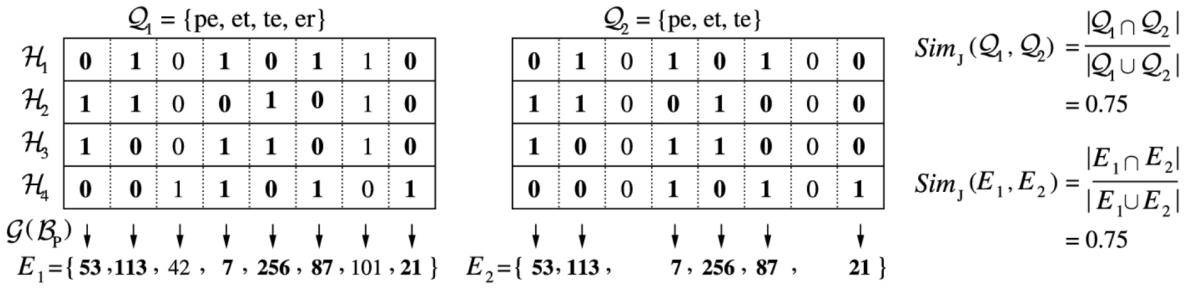

Figure 2.4.: Example of Two-Step Hashing on the Input Strings 'peter' and 'pete', Source: [RCS20], p. 141

2.4. Similarity Graph Creation

When all records in D_{private} and D_{public} have been encoded, the respective similarity graphs can be created. This is the first step that is not generally used in PPRL, but is rather necessary to execute a GMA afterwards.

The similarity graph creation still happens independently from each other for the encoded data of D_{private} and D_{public}. The set of nodes N_{sim} of the similarity graph G_{sim} consists of all encoded records in D, and the nodes can be distinguished and referred to by, e.g., their respective IDs. The edges E_{sim} of G_{sim} can now be determined by calculating the pairwise distances between nodes using metrics like the Dice of Jaccard coefficient. The resulting graphs $G_{\text{sim, private}} = (N_{\text{sim, private}}, E_{\text{sim, private}})$ and $G_{\text{sim, public}} = (N_{\text{sim, public}}, E_{\text{sim, public}})$ represent the pairwise similarities of all entries within D_{private} and D_{public} respectively.

2.5. Embedding through Node2Vec Reduction

The embedding step aims at making the neighbourhood structures of $G_{\text{sim, private}}$ and $G_{\text{sim, public}}$ comparable with each other. So far, only similarities of nodes within one graph have been considered. However, to allow for RL across the two datasets, this is not sufficient anymore. Rather, the structure of the neighbourhood of a node is used to look for a node with a very similar neighbourhood in the other similarity graph. Consequently, these two nodes most likely encode records that refer to the same individual.

Node2Vec is an embedding algorithm introduced by Grover and Leskovec [GL16] and is based on the Word2Vec algorithm [Mik+13] which learns word embeddings within sentences to capture semantic meanings and relationships between words. It preserves the overall graph topology, thus in particular the neighbourhood structures of every node, and has been proven to be successful for related tasks in other domains [Pal+18; PGS19].

The Node2Vec algorithm consists of two main steps: random walks through the similarity graph G_{sim} and the application of Word2Vec. First, Node2Vec performs a set number of random walks from every node in G_{sim} to explore the graph structure. This works as follows: the two hyperparameters p and q determine whether the random walks tend to explore the graph more in depth or width. The return parameter p controls the likelihood to directly return to the previous node. A high value makes it less likely to return, encouraging a broader exploration of the graph. q is called the in-out parameter and determines the likelihood to visit nodes that are close to or far from the previous node. A high value makes it more likely

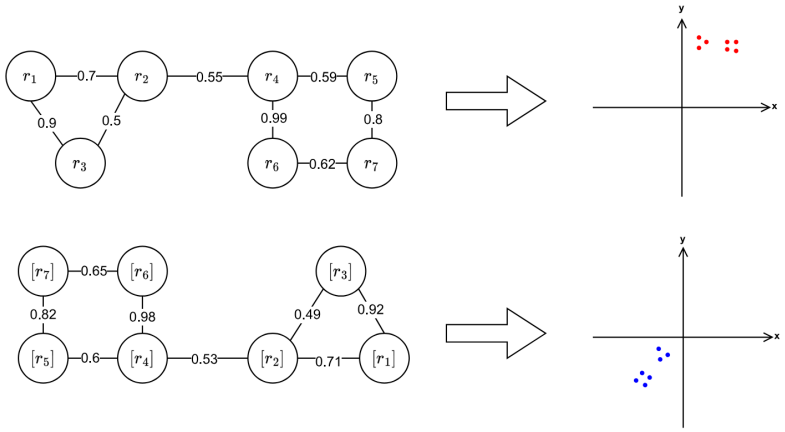

Figure 2.5.: Creation of Embeddings for Nodes in $G_{\text{sim, private}}$ and $G_{\text{sim, public}}$ in Similar Neighbourhoods, Source: [SA24], p. 6

to visit nodes further away, simulating a Depth-First Search (DFS)-like behavior, while a low value makes it more likely to visit close nodes, simulating a Breadth-First Search (BFS)-like behavior.

Every random walk creates a sequence of nodes visited. These sequences are subsequently interpreted as sentences, and the single nodes in one random walk as words. In the second step, the Word2Vec algorithm is then being executed on these sentences as its input. By applying Word2Vec, the nodes of G_{sim} are transformed into vectors in a low-dimensional feature space, so-called embeddings [GL16], that still preserve the neighbourhood structures of G_{sim} as accurately as possible. This can be done using Skip-gram training, effectively maximizing the probability to correctly predict the surrounding nodes within a fixed range given a target node n. For this, Word2Vec takes a fixed context window of size s and learns to predict the surrounding s words of a given target word. As the sentences from the random walks through $G_{\text{sim, private}}$ and $G_{\text{sim, public}}$ are used for the training of the embeddings emb_{private} and emb_{public}, and the random walks directly depend on the neighbourhood structures of the two graphs, one can expect to receive similar embeddings in emb_{private} and emb_{public} for nodes sharing similar graph neighbourhoods.

Fig. 2.5 demonstrates how embeddings, i.e., vectors that can be located in a graph, are created from $G_{\text{sim, private}}$ and $G_{\text{sim, public}}$. Note that although the two graphs show very similar neighbourhood structures for their nodes, their respective embeddings do not necessarily need to have the same or similar vector representations. Instead, the embeddings only have comparable distances to their respective neighbouring nodes. However, this allows for a comparison of neighbourhoods in the next step to finally match nodes across the two graphs that most likely refer to the same individual.

2.6. Embedding Alignment through Wasserstein Procrustes

After the creation of node embeddings for the two similarity graphs, the alignment step aims at transforming one embedding as closely into the space of the other one as possible by applying translation, scaling, and rotation. The two embeddings are matrices of the form $emb_{\text{private}} \in$

$\mathbb{R}^{|\mathcal{D}_{\text{private}}| \times d}$ and $emb_{\text{public}} \in \mathbb{R}^{|\mathcal{D}_{\text{public}}| \times d}$. That means that emb_{private} and emb_{public} are matrices that contain one embedding for one particular record in D_{private} or D_{public} in every of their $|D_{\text{private}}|$ or $|D_{\text{public}}|$ rows respectively, every embedding consisting of d real numbers.

To identify which rows in emb_{private} and emb_{public} likely refer to the same individual, one embedding needs to be aligned into the space of the other one to allow for bipartite graph matching in the following step. Without loss of generality, assume that the matrix emb_{private} should be aligned to emb_{public}. For that, two measures must be achieved during the embedding alignment step: First, emb_{private} must be transformed, i.e., translated, scaled and rotated, so that the distances between corresponding embeddings $emb_{\text{private}}[i]$ and $emb_{\text{public}}[j]$ are as small as possible. And secondly, it has to be decided which embeddings from emb_{private} and emb_{public} actually belong together, i.e., whose referring records identify the same individual.

For the first problem, a Procrustes analysis can be performed [1]. This computes a transformation matrix Q^* such that the difference between $emb_{\text{private}} \times Q^*$ and emb_{public} is minimized. Or more formally, one computes a solution for the following optimization problem:

$$\min_{Q \in O_{\text{d}}} ||emb_{\text{private}} \times Q - emb_{\text{public}}||_F^2$$

where F refers to the Frobenius norm of a matrix and Q_{d} to the set of orthogonal matrices of size $d \times d$ [AP21; GJB19]. For a matrix M to be orthogonal, it must apply that $M^T \times M = I$ where I is the identity matrix and M^T is M transposed. The squared Frobenius norm $||M||_F^2$ represents the sum of the squared differences between the elements of the matrix M and is defined as:

$$||M||_F^2 = \sum_{i,j} (|M_{ij}|^2)$$

To be able to assign the correct embeddings in emb_{private} and emb_{public} to each other, the squared Wasserstein distance can be used. It is sometimes referred to as the earth mover's distance [RTG98]. It is a measure of similarity between two frequency distributions. Informally, this distance describes the minimum cost of transforming one pile of earth, or dirt, into another one, both being described by the respective distribution. The cost is measured in the amount of dirt moved, times the distance over which it is moved. Formally, in this second step, the following optimization problem is being solved:

$$\min_{P \in \mathcal{P}_n} ||emb_{\text{private}} - P \times emb_{\text{public}}||_F^2$$

where $\mathcal{P}_n$ is the set of permutation matrices guaranteeing a 1-1 mapping between emb_{private} and emb_{public} [GJB19].

For Wasserstein Procrustes, neither the corresponding nodes nor the transformation matrix need to be known at the beginning. Instead, over multiple epochs containing multiple iterations, an increasingly better $Q \in O_d$ is computed such that the embeddings of emb_{private} are close to the embeddings in emb_{public}, and a 1-1 correspondence can be derived [GJB19].

When the final transformation matrix Q^* has been calculated, the source matrix emb_{private} can be multiplied with Q^* to project its embeddings into the space of emb_{public}, i.e., $emb_{\text{private, projected}} = emb_{\text{private}} \times Q^*$.

This process of projecting one embedding into the space of the other one is visualized in Fig. 2.6. Here, the set of blue dots represents emb_{private}, and the set of red dots emb_{public}. By

[1] "In ancient Greek, Procrustes' name referred to a bandit who tortured his guests to make a perfect fit with his bed by stretching their limbs or cutting them off", Source: [AP21], p. 77

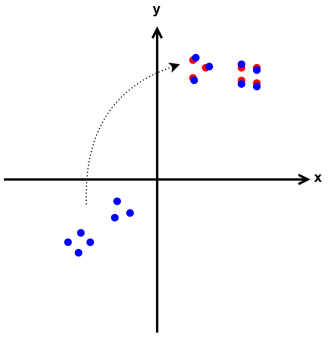

Figure 2.6.: Alignment of the Embeddings of $G_{\text{sim, private}}$ Into the Space of Embeddings of $G_{\text{sim, public}}$, Source: [SA24], p. 6

applying Q^* to emb_{private}, these embeddings can be transformed very closely to the embeddings in emb_{public}.

2.7. Graph Matching

Given $emb_{\text{private, projected}}$ and emb_{public}, the last step of GMA_{ML} consists of matching those records in D_{private} and D_{public} together whose embeddings are highly similar to each other. To measure the similarity of an embedding pair $emb_{\text{private, projected}}[i]$ and $emb_{\text{public}}[j], 1 \leq i \leq |D_{\text{private}}|, 1 \leq j \leq |D_{\text{public}}|$, the pairwise similarities of $emb_{\text{private, projected}}[i]$ and $emb_{\text{public}}[j]$ can be calculated using a metric in the Euclidean space.

A widespread and well suitable metric for our application is the cosine similarity metric. It indicates the cosine of the angle between two vectors, without considering the lengths of the vectors. This makes the cosine similarity metric more robust against very stretched and differently scaled embeddings [SA24]. For two vectors $\vec{u}$ and $\vec{v}$ of the same dimension, their cosine similarity can be calculated as follows:

$$cos(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{||\vec{u}|| \cdot ||\vec{v}||}$$

where $|| \cdot ||$ denotes the L^2 or Euclidean norm of a vector [LH13].

For the set of edges between the embeddings

$$sim_{embeddings} = \{(emb_{\text{private, projected}}[i], emb_{\text{public}}[j]),\ \cos(emb_{\text{private, projected}}[i], emb_{\text{public}}[j]) \mid 1 \leq i \leq |D_{\text{private}}|,\ 1 \leq j \leq |D_{\text{public}}|\}$$

that contains the similarities of all possible record matches between D_{private} and D_{public}, a matching between the two record sets can then be performed based on the respective similarities between the embeddings of two records.

Bipartite graph matching algorithms can therefore be applied to match as many pairs as possible, while at the same time maximizing the probability that two matched records in fact refer to the same individual.

Thus, two objectives are being pursued at the same time, each with a different importance for different applications: First, as many pairs of records must be identified. These pairs then represent the final output of GMA_{ML}, identifying which records in D_{private} and D_{public} refer to the same individuals. When the primary goal is to simply re-identify as many records in D_{private} as possible, this number of matched pairs must be maximized.

If a certain RL quality is required, a second objective must be dealt with: In this case, additionally to the first goal of identifying as many matching record pairs as possible, the number of falsely positively matched records must be minimized as well. For a malicious actor for example, it might be less important to ensure a high RL quality. Instead, their main objective is a high re-identification rate. If RL is performed in its original, non-malicious form however, the two database owners O_1 and O_2 likely want to be sure that the linked records have mostly been identified correctly and are thereby reliable and useful for further applications. In this case, a trade-of between a tolerable number of false positives and a high number of linked records in total must be agreed on.

For these two main objectives, there are different graph matching techniques ensuring certain properties. The problem of selecting the edges of the most similar pairs in $sim_{\text{embeddings}}$ while ensuring that every record in D_{private} is only linked to at most one record in D_{public} is a bipartite graph matching problem [TIR78]. That is, the graph for which the matching is calculated, looks as follows: $G_{\text{bipartite}} = (emb_{\text{private, projected}} \cup emb_{\text{public}}, sim_{\text{embeddings}})$. The nodes of $G_{\text{bipartite}}$ are all nodes representing the records in D_{private} and D_{public}, and the weighted, undirected edges only connect nodes from one set $emb_{\text{private, projected}}$ to the other one emb_{public} with the two sets being disjoint, i.e., $emb_{\text{private, projected}} \cap emb_{\text{public}} = \emptyset$.

For $G_{\text{bipartite}}$, different bipartite graph matching algorithms can be applied, like minimum weight matching, matching by solving the stable marriage problem, symmetric graph matching or NN matching. They guarantee different matching properties and thus are differently advantageous for scenarios with different main objectives.

Minimum Weight Bipartite Graph Matching The minimum weight bipartite graph matching finds a full 1-1 mapping between $emb_{\text{private, projected}}$ and emb_{public} in a way that the overall distance between paired nodes in minimized. Given the weight function w that reflects the cosine similarities of two embeddings, the pairs with similarity scores are selected such that:

$$min(\sum_{e \in sim_{\text{embeddings}}} w(e)x_e)$$

where x_e is a binary value indicating if e is included in that particular matching.

For $|emb_{\text{private, projected}}| = |emb_{\text{public}}|$, a bijective mapping from one set of nodes to the other one is created. If the sets are not of equal size, every node in the smaller set is matched to exactly one node in the larger set [BV10; Cro16].

Matching Through the Stable Marriage Problem By solving the stable marriage problem, a full 1-1 mapping can be created as well. The problem can be formulated as follows:

> "In a group of n men and n women, each person ranks the members of opposite sex as potential marriage partners. A matching (marriage) M between the set of men and the set of women is called stable if there is no pair (m, w) of a man m and a woman w who are not matched but prefer each other to their partners in the matching." [Pit92], p. 358

Thus, for the two sets $emb_{\text{private, projected}}$ and emb_{public}, a stable matching can be calculated where every embedding ranks the embeddings from the opposite dataset by their cosine similarity.

Symmetric Graph Matching Symmetric graph matching computes a 1-1 matching in such a way that two embeddings $emb_{\text{private, projected}}[i]$ and $emb_{\text{public}}[j]$ are matched together if and only if $cos(emb_{\text{private, projected}}[i], emb_{\text{public}}[j])$ provides the highest similarity value for $emb_{\text{private, projected}}[i]$ to any other node in emb_{public}, and vice versa.

Thus, unlike stable and minimum weight matching, symmetric graph matching does not guarantee a full mapping. This might reduce the total number of re-identified records, but it can also lower the false positives rate as both records have to be the most similar ones to each other to actually be mapped together.

Nearest Neighbour NN matching chooses the embedding $emb_{\text{public}}[j] \in emb_{\text{public}}$ for every $emb_{\text{private, projected}}[i] \in emb_{\text{private, projected}}$ that is the most similar to $emb_{\text{private, projected}}[i]$ and matches these two records together, i.e., for $emb_{\text{private, projected}}[i]$, an $emb_{\text{public}}[j]$ is chosen such that

$$emb_{\text{public}}[j] = min_{emb_{\text{public}}[k] \in emb_{\text{public}}}(cos(emb_{\text{private, projected}}[i], emb_{\text{public}}[k])$$

This mapping approach neither guarantees a full, nor a bijective 1-1 mapping as embeddings in emb_{public} can occur in matched pairs multiple times, or not at all [EPY97]. At the same time, it is significantly more efficient than 1-1 mappings, especially for increasingly larger datasets, and since it does not force a full mapping, it can reduce the number of falsely positively matched record pairs.

3. Related Work

The experiments conducted for this thesis are based on the GMA introduced in [SA24]. Thus, this chapter summarizes significant contributions in the area of GMAs in recent years with a focus on the advancements achieved by [SA24].

Other attack strategies outside of the realm of GMAs mostly focus on exploiting weaknesses within the used encoding mechanisms. This has been proven to be successful for BFs as they are vulnerable to dictionary attacks or cryptanalysis [Mit+17; Vid+19]. However, more recently proposed schemes such as BFs with diffusion, TMH or TSH have addressed these weaknesses and are therefore considered to be secure against privacy attacks. However, as GMAs exploit the inherent need of PPRL to be able to calculate similarities between records, if encoded or not, they have emerged as the most prevalent threat for PPRL [SA24].

Culname et al. In 2017, Culname et al. [CRT17] have shown a first effective GMA as part of their research on potential weaknesses of the PPRL practices of the UK Office for National Statistics (ONS). Amongst others, they were able to create a similarity graph for the plain-text data as the ONS has published similarity tables for their not yet encoded data together with the encoded data itself. Assuming a subset of names stored in the ONS database is already known to the attacker, they can then find a graph isomorphism between the subgraph of known names and the complete graph. In their specific use case, and for an overlap of 100%, they were able to re-identify 93% of all records, and this success rate can be kept at 55% for an overlap of only 30%. This is though, assuming that $D_{\text{private}} \subseteq D_{\text{public}}$.

In their work, Culname et al. have already recommended to avoid sharing similarity tables of plain-text records as this enabled them to construct a graph of the dataset used by the ONS. Additionally, in their attack they assume that the weighted edges of the plain-text graph and the encoded one have identical values. These are assumptions that will likely not be given in future real-world applications, limiting the effectiveness of the GMA for other applications apart from their case.

Vidanage et al. A more advanced GMA has been proposed by Vidanage et al. [Vid+20] whose general steps and ideas are already very similar to the GMA used in this thesis. They generate two similarity graphs from a publicly available plain-text dataset, and an encoded one. Afterwards, the nodes are represented by several features such as node frequency, length and degree centrality, and a feature vector for every node is calculated to allow for distance calculations in Euclidean space. Lastly, bipartite graph matching between the two graphs of feature vectors is performed using, amongst others, matching through solving the stable marriage problem, and minimum weight matching. For an overlap of 100%, a re-identification rate of up to 100% has been achieved.

Although this seems to be very promising, the GMA makes a number of assumptions that question its application on real-world scenarios: First, the attack only works for very high overlaps. A drop of the overlap value by only five percentage points to 95% already reduces the re-identification rate to 0.5% [Hen+22]. Secondly, some of the conducted experiments

only showed satisfactory results if a pre-processing step is done prior to the feature extraction. For this, a ground truth of some pairs of encoded and plain-text data that refer to the same individual must be known by the attacker beforehand. This ground truth is then used to adjust the similarities in both graphs before applying the feature vector calculation. However, if such a ground truth is given in other cases is highly questionable.

Lastly, [SA24] have discovered a restriction in the GMA of [Vid+20]: In their pre-processing blocking step, [Vid+20] use the same random seed for both, the private and the public dataset, to split the records into different blocks. Usually, this seed would be expected to be kept secret by the data owners that want to perform PPRL on their data. As this blocking step is crucial to high re-identification rates, it is questionable how effective this GMA would be in scenarios with a more realistic attacker knowledge.

Heng et al. In their study on the effectiveness of GMAs against PPRL, Heng et al. have shown that unlike previous papers suggested, no GMA posed a significant threat to common PPRL techniques at that time [Hen+22]. The main reason for that is that all considered GMAs assume a very high overlap rate of D_{private} and D_{public} of mostly 95% or higher. Though, with decreasing overlap rates, the success rates indicating how many records from both datasets were able to be matched correctly, drop sharply. This makes GMAs very unsuccessful for most applications. Additionally, even if the attacker can access a public dataset with almost perfect overlap to D_{private}, [Hen+22] proposed a simple, yet effective countermeasure: By inserting fake records into D_{private}, the overlap rate can be reduced reliably, resulting in a complete undermining of all GMAs relying on high overlap rates.

Armknecht et al. In previous work, BFs have been proven to be susceptible to frequency and graph matching attacks [Chr+18; Vid+19]. Thus, Armknecht et al. [AHS23] propose to add a linear diffusion layer to the BF encoding process to break the statistical relationship between the q-gram inputs and the resulting output bits. This weakens the correlation between the similarity of plain-text and encoded records that refer to the same individual, thus lowering the success rate of GMAs. At the same time though, [AHS23] showed that this added layer of diffusion does not significantly reduce the linkage quality for an application on PPRL compared to traditional PPRL with BFs.

Schäfer et al. On the way towards a universal GMA requiring minimal attacker knowledge, Schäfer et al. were able to provide a significant contribution in form of an unsupervised ML GMA [SA24]. It uses the techniques and mathematical foundations explained in chapter 2. The newly proposed combination of the Node2Vec algorithm for proper node embedding, and the subsequent unsupervised Wasserstein Procrustes analysis work very well to bring those vectors across D_{private} and D_{public} close together in the Euclidean space that refer to the same individual. The final step of bipartite graph matching using the cosine similarity metric and minimum weight matching then results in high linkage success rates, outperforming any other GMA proposed so far. This also applies to relatively low overlaps of 25% and lower and to the case that $D_{\text{private}} \not\subseteq D_{\text{public}}$, resulting in a significantly more realistic attacker model with minimal knowledge required. Thus, this GMA by [SA24] is the first of its kind that poses a realistic threat to commonly used PPRL techniques, making an attack realistic for real world scenarios.

Nonetheless, the attack has been proven to still being ineffective against BFs with diffusion

as proposed by [AHS23]. Additionally, only a full bipartite graph matching has been evaluated so far in their experiments. This however forces every record in D_{private} to be matched with exactly one record in D_{public}, assuming that $|D_{\text{private}}| \leq |D_{\text{public}}|$. This results in a potentially high number of falsely positively matched records, especially for small overlaps and linkage success rates. Thus, although the GMA by [SA24] has been proven to overcome most of the limitations of previous GMAs, there is still room for improvement, as well as further examinations to gain deeper insights in the properties of the attack.

4. Experimental Setup

The work conducted in this thesis is based on the unsupervised ML GMA by [SA24], GMA_{ML}, and aims at exploring certain properties of the algorithm more deeply in order to gain a better understanding of possible improvements as well as its limitations. Additionally, the experiments carried out aim at evaluating a possible application of GMA_{ML} for PPRL in its original, non-malicious form.

In order to achieve this, four different experimental setups were created, examining different real world use cases and properties of GMA_{ML}. Their applications, setups and implementations will be explained in detail in the following sections.

As this work additionally examines use cases in the domain of PPRL rather than GMAs exclusively, unlike the work in [SA24], the secret salts used by the two data owners O_1 and O_2 in the data encoding step are set to the same value. This represents a realistic real world scenario as in the interest of maximizing the linkage success, O_1 and O_2 likely communicate the secret salt beforehand via a secure communication channel. However, if not stated otherwise, all other parameters set in [SA24] are kept the same.

4.1. Datasets and Computational Power

The datasets on which the following experiments have been executed are part of those already used by [SA24] allowing for not only a well-founded comparison of experimental results within this thesis, but also with the base case experiment run in [SA24].

However, due to limited computational resources, the two largest datasets with 20,000 and 50,000 records have not been considered for most of the experiments. Instead, the datasets consisting of 1000, 2000, 5000 and 10,000 fake names each have been used, as well as the widely used Titanic dataset[1]. The fake name datasets have been created using a fake name generator[2] to obtain random first and last names, dates of birth and user IDs. From the Titanic dataset, only the first and last name and an ID have been selected for the 891 record entries.

In the following experiments, the first and last names together with the date of birth form the PIIs that always needs to be encoded to create experiments that can be transferred to real-world use cases.

For all experiments, the linkage success has been evaluated for different overlapping sizes o of the two datasets D_{private} and D_{public} from 5% to 100%, i.e., $o \in \{0.05, 0.10, 0.15, \ldots, 1\}$. This allows for every variation of GMA_{ML} to evaluate the minimal overlapping of D_{private} and D_{public} required to achieve a desired or meaningful linkage success rate (LSR).

Again, due to limited computational resources, the case that $D_{\text{private}} \subseteq D_{\text{public}}$, i.e., that every record from the private, encoded dataset can also be found in the publicly available dataset, has not been considered. Rather, the experiments focus on the scenario where

[1] available under, amongst others, https://github.com/datasciencedojo/datasets/blob/master/titanic.csv
[2] https://www.fakenamegenerator.com

$D_{\text{private}} \nsubseteq D_{\text{public}} \wedge D_{\text{public}} \nsubseteq D_{\text{private}}$, i.e., some records in D_{private} have no equivalent in D_{public} and vice versa.

All experiments were carried out in a virtual machine running Ubuntu 22.04 on 20 cores of an AMD EPYC 7272 CPU with access to 180 GB of DDR-4 memory. Additionally, the virtual machine had access to an NVIDIA GeForce RTX 3090Ti graphics card with 24GB of VRAM.

4.2. Enhancing Linkage Success through Ground Truth

The first experiment implemented as part of this thesis examines the influence of a given ground truth on the LSR. Please recall that it is intrinsic to PPRL that only those individuals can be matched successfully who have record representations in D_{private} and D_{public}. The LSR thus determines how many of these potentially linkable records have in fact been linked together successfully and can be calculated as follows:

$$\text{LSR} = \frac{\#\text{ correct matches}}{|D_{\text{private}} \cap D_{\text{public}}|}$$

So far, the results from [SA24] show that their GMA performs increasingly well for increasingly larger datasets and overlaps of D_{private} and D_{public}. This has in fact been expected since both, greater overlaps and larger datasets provide more linkable record pairs and thereby also improved performances of the node embedding and alignment steps of GMA_{ML}. However, this makes it difficult for data owners of smaller datasets or smaller overlap rates to successfully link those few records that refer the same individuals nonetheless.

One possible solution for these use cases could be the selection of a common ground truth before performing the embedding alignment step in GMA_{ML}. The ground truth consists of all pairs of encoded records $(r_{\text{enc, private}}, r_{\text{enc, public}}), r_{\text{enc, private}} \in D_{\text{enc, private}} \wedge r_{\text{enc, public}} \in D_{\text{enc, public}}$ whose encodings match exactly. Thus, the two records from the different datasets refer to the same individual. By selecting only the common ground truth records for the embedding alignment step, one can prevent records that do not have a matching record in the other database to make the resulting transformation matrix more inaccurate. Nonetheless, at the same time, this also reduces the data used for aligning the two graph embeddings, potentially worsening the LSR even more, especially for small datasets or overlaps.

The experiments will show if this additional step of ground truth selection does in fact enhance the LSRs, or if additional measures have to be taken into account. For this first experiment, the LSR is evaluated for the datasets of 1000, 2000, 5000 and 10,000 fake names with the standard overlaps from 5% to 100%.

In the implementation of GMA_{ML}, after the node embedding using Node2Vec and before aligning the embeddings applying a Wasserstein Procrustes analysis, the ground truth is selected out of the two embeddings emb_{private} and emb_{public}. To successfully filter the embeddings that represent records that have a matching partner record in the other dataset, one has to make sure that the order of the embeddings is still relatable or traceable back to the order of the encodings enc_{private} and enc_{public}. This is necessary in order to select exactly those embeddings whose related encodings are equal to each other across the two datasets. This can either be achieved by not changing the order of records at all from the encoding step to the ground truth selection, or by associating the user IDs directly to the data at respective times.

As the embedding step in the GMA_{ML} originally permutes the order of the nodes randomly before returning the respective embeddings and their user IDs, the first case is not given

initially. Therefore, the embeddings *emb* that have been shuffled randomly and are thereby ordered by their IDs_{emb} have to be sorted by their original order of IDs_{enc}. Afterwards, emb_{private} and emb_{public} are ordered the same way as enc_{private} and enc_{public} respectively again.

Now, enc_{private} and enc_{public} can be compared entry-wise, and when it holds that $enc_{\text{private}}[i] = enc_{\text{public}}[j]$, $emb_{\text{private}}[i]$ and $emb_{\text{public}}[j]$ are added to the ground truth sets GT_{private} and GT_{public} for $1 \leq i \leq |D_{\text{private}}|, 1 \leq j \leq |D_{\text{public}}|$.

Afterwards, instead of the whole sets of embeddings emb_{private} and emb_{public}, only GT_{private} and GT_{public}, i.e., subsets of the whole embeddings, are used as input for the following embedding alignment step.

4.2.1. Ground Truth with Dummy Dataset

One possibly occurring problem of the ground truth selection is that for small datasets or overlaps, the ground truth might be too small to enhance the LSR, or might even worsen it. One solution to avoid this problem might be to add a dummy dataset to the original datasets D_{private} and D_{public}. This would increase the number of embeddings selected for ground truth, likely improving the results of the embedding alignment and thus, resulting in a higher LSR.

To examine the effectiveness of ground truth selection with a dummy dataset, an additional dataset of 5000 fake names has been generated using the fake name generator. For maximum adoption to the other used datasets, every fake individual is once again stored with first and last name, date of birth and a user ID. As later on, the records can only be identified by their respective IDs, and to avoid confusion or ambiguity, the IDs in the dummy dataset D_{dummy} must not overlap with any IDs from all other datasets used for this experiment. Consequently, the datasets of 1000, 2000, 5000 and 10,000 fake names are used, and for every dataset, overlaps from 5% to 100% are evaluated once again. Additionally, for each of these combinations of dataset and overlap, a dummy set of 10, 100, 1000, 2000 or 5000 fake names each is added to D_{private} and D_{public} prior to the encoding step.

The encoding and node embedding are then performed as usual on these newly formed datasets. After the nodes have been embedded, the ground truth is selected as described above, and the Wasserstein Procrustes analysis is performed on the ground truth. Before the embeddings of D_{private} and D_{public} are being matched together using bipartite graph matching, the dummy values have to be sorted out to avoid matching a dummy value with a non-dummy one, potentially lowering the LSR unnecessarily. Afterwards, the LSR can be calculated as indicated above.

4.3. Comparing Bipartite Graph Matching Techniques for an Optimal Trade-Off between Linkage Success and False Positives Rate

The second experiment conducted within this thesis is based on a different evaluation of success for PPRL and GMAs: GMAs aim to re-identify as many individuals as possible. For that, a high rate of false positives can be negligible, i.e., it does not primarily matter if records, that do not actually refer to the same individual, are linked, as long as the number of correct matches itself is maximized. This principle has also been applied in [SA24], where the application of minimum weight bipartite graph matching creates a full mapping, meaning that every record in D_{private} is matched to one record in D_{public}. This applies for $|D_{\text{private}}| \leq |D_{\text{public}}|$. Otherwise, $|D_{\text{public}}|$ records in D_{private} are matched, i.e., the maximum number possible.

In PPRL however, the goal is not only to match as many records as possible, but to also guarantee linkage quality to a certain degree. Thus, additionally, the false positives rate (FPR) must be kept as low as possible. In order to achieve this, a full 1-1 matching as in minimum weight matching is not optimal in every use case, especially not for small datasets or overlaps where the LSR often is quite low [SA24].

Therefore, this experiment focuses on comparing four different bipartite graph matching techniques: the above mentioned minimum weight matching, graph matching through solving the stable marriage problem, symmetric matching and NN matching. While minimum weight and stable marriage matching always create full 1-1 matchings for $|D_{\text{private}}| = |D_{\text{public}}|$, symmetric matching does not guarantee a full matching, and NN matching does not guarantee 1-1 matchings.

To compare the different graph matching techniques, the LSR for these four matching approaches is evaluated on the datasets of 1000, 2000, 5000 and 10,000 fake names as well as the Titanic dataset for dataset overlaps of D_{private} and D_{public} from 5% to 100%.

Additionally, the FPR is measured. It can be calculated as:

$$\text{FPR} = \frac{\#\text{ wrong matches}}{\#\text{ matched records}}$$

A combined evaluation of the respective LSRs and FPRs will show which graph matching technique is best suitable and provides the best trade-off between a high LSR and a low FPR.

4.4. Runtime Behaviour for Increasingly Larger Datasets

An asymptotic runtime analysis has already been conducted in [SA24]. They have concluded that the time complexity of the whole GMA_{ML} is in $O(n^3)$, n equals the size of the larger dataset, for encoding using BFs, TMH or TSH, node embedding through Node2Vec, embedding alignment performing a Wasserstein Procrustes analysis and minimum weight bipartite graph matching. The highest time complexity has bipartite graph matching as the underlying Jonker-Volgenant algorithm used in their implementation [Cro16] is of cubic runtime [SA24].

While a theoretical runtime analysis can be helpful for a first impression of the actual runtime, it is still of great interest for possible real-world applications of GMA_{ML} or variants of it to measure its runtime behaviour for different sizes of input datasets. Thus, this third experiment aims at getting a first impression and evaluating the total runtime of GMA_{ML} for the datasets with 1000, 2000, 5000, 10,000, 20,000 and 50,000 fake names and overlaps from 5% to 100%. Additionally, an analysis and comparison of the runtimes of the four major steps in GMA_{ML} will show what steps, despite the theoretical runtime analysis, actually provide the largest time constraints and limitations to the whole algorithm.

4.5. Record Linkage Success for Increasingly Erroneous Datasets

So far, in this thesis and other work like [SA24], only the case has been considered where the overlapping values in D_{private} and D_{public} are exactly equal. However, this is not likely to be applicable in real-world applications for two reasons: First, in the case of a GMA, the malicious attacker does not know what PII the data owner O has used as input for the encoding of the sensitive data. While the problems that arise for the attacker with this uncertainty can be minimized by executing the attack multiple times, each time with a different subset of PII that

could have been used by O, a second cause of uncertainty cannot be eliminated: The data stored in the datasets used in the process of either the GMA or PPRL can be erroneous for several reasons. Most prevalent are probably different naming conventions and data formats as well as spelling or typing errors.

Problems in naming conventions can occur in many attributes of PII. In last names for example, where suffixes like "junior" or "senior" might be abbreviated differently, e.g., "Jr." in contrast to "Jr" or "Sr." in contrast to "Sr". Also, addresses can be subject to orthographic variations where, e.g., "Sheev Street" and "Sheev St." might refer to the identical street but are spelled differently. Different data formats, mostly occurring in the use of dates and times like for dates of birth, can also lead to different spellings, and thus different encodings for the same entities in different datasets. Dates of birth are most likely casted to strings before encoding them, therefore being represented in the same data type. However, different date formats like dd/mm/yyyy or yyyy/mm/dd also create different strings for the same date.

Lastly, and least avoidable, spelling and typing errors can occur when entries are added to or edited in a dataset. These inaccuracies often probably only lead to small differences between entries referencing the same person in different datasets. However, they lead to slightly different encodings of records nonetheless.

Because of these potentially and partly unavoidably erroneous record entries, it must also be considered how well GMA_{ML} performs on these data. Therefore, the last experiment conducted as part of this thesis aims to evaluate how well GMA_{ML} works for increasingly erroneous data. The original datasets of 1000, 2000, 5000 and 10000 fake names are taken, and their PII, i.e., first and last name as well as date of birth, are falsified to a certain percentage. To compare the quality of the whole linkage process based on how erroneous one dataset is, the error rate ranges from 2% to 20%, with rates of 2%, 5%, 10%, 15% and 20%. A higher error rate has not been considered due to the already high computational costs of conducting experiments on 5 base datasets for 20 different overlaps each, and for 5 different error rates for every base dataset. Additionally, in most datasets available and used in real-world contexts, a consistent error rate of over 20% for the whole dataset is very unlikely, or these datasets would not be considered for PPRL without prior data cleansing.

To include errors in the records of the respective datasets, these datasets have been pre-processed before applying GMA_{ML} on them: For every record in the dataset, every independent piece of PII is taken separately. For every character within that string, with the specified error rate probability, that character is changed to a random character, digit or punctuation mark. This ensures that the expected error rate of the resulting dataset is the previously specified rate.

Lastly, GMA_{ML} has to be altered in such a way that it takes two different datasets as input: the original and the erroneous one with the specified error rate. Originally, GMA_{ML} only takes one dataset as input, and this dataset is split into records exclusively in D_{private}, exclusively in D_{public}, and present in both D_{private} and D_{public}. The crucial point in this experiment however is that even the records in the overlap between D_{private} and D_{public} can be different. They still refer to the same individual and thus should me matched together later on, but due to the errors in one dataset, they potentially do not match exactly anymore. Thus, D_{private} and D_{public} originate from two different datasets: the base dataset of fake names, and its erroneous pendant with the specified error rate. The overlapping values, i.e., the records present in both D_{private} and D_{public}, are not in the exact same set of records anymore. Rather, it is made sure that the user IDs in the base and erroneous datasets match for the same individual, and then based on a selected number of indices, these corresponding records are then taken as the

overlapping values. The remaining records are split evenly among D_{private} and D_{public}.

5. Experimental Results and Evaluation

The main indicator for the quality of RL or GMA algorithms is their linkage success rate (LSR) that indicates how many records in $D_{\text{private}} \cap D_{\text{public}}$ have been matched together successfully. Please recall that only those records can be matched successfully that are present in both datasets, thus the formula for the LSR introduced in section 4.2. For the experiments that aim at evaluating the effect of certain measures, like using a known ground truth for the embedding alignment, or evaluating the influence of errors in datasets, the LSR is examined depending on the record overlap of both datasets.

Additionally, a metric can be introduced called the minimum required overlap (MRO) to specifically measure how successful a test run has been for different scenarios. It indicates the smallest overlap for which 50% or more of the possibly re-identifiable records have actually been matched together. As smaller overlaps require less efforts from the data owners in case of RL or from an attacker in case of a GMA, lower MROs indicate a stronger RL or GMA algorithm.

Although the LSR stays the most important measure of success, the false positives rate (FPR) is taken into account as well for the comparison of different bipartite graph matching techniques. Therefore, the FPR is evaluated depending on the respective overlaps to compare the linkage quality of the different graph matching techniques used.

Lastly, the third experiment conducted aims at evaluating the runtime of the algorithm for different dataset sizes. Thus, the total runtime is measured for every dataset as well as the runtimes for every major step of GMA_{ML}.

5.1. Analysis of Conducted Experiments

To confirm the results obtained in [SA24], and as a basis for the further experiments, two base case runs of GMA_{ML} have been executed using BF encoding and minimum weight graph matching. The first run considers the optimal case that $D_{\text{private}} \subseteq D_{\text{public}}$, and the second one the more realistic scenario where $D_{\text{private}} \nsubseteq D_{\text{public}}$. The only difference in this run of GMA_{ML} is that the secret salt used to encode the records in D_{private} and D_{public} is set to the same value. While [SA24] focus on an application of their algorithm in the realm of GMAs, this thesis also examines the application of GMA_{ML} on RL in its non-malicious form. For this reason, we can assume that O_1 and O_2 have agreed on a salt for the encoding step in advance over a secure channel. Thus, the success rates in our initial test runs are expected to be slightly better due to the similar salt value, but the overall tendencies should be similar.

A detailed figure and the corresponding table summarizing the LSRs and MROs for $D_{\text{private}} \subseteq D_{\text{public}}$ can be found in the appendix A.1.1. For the base case and $D_{\text{private}} \nsubseteq D_{\text{public}}$, an analysis of the MRO as defined earlier shows that the run with identical salts produces slightly better results. In general however, the base case experiment fully confirms the results from [SA24]: The overall tendencies are equal, larger datasets reach higher LSRs earlier, and for every dataset, there seems to be one critical overlap value that significantly increases the LSR.

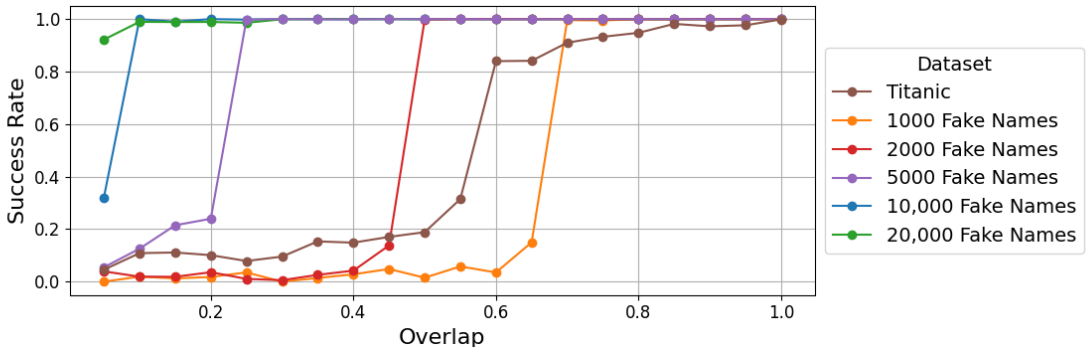

Figure 5.1.: Success Rates for the Base Case Scenario, and $D_{\text{private}} \not\subseteq D_{\text{public}}$.

Datasets	Titanic	1000	2000	5000	10,000
[SA24]	0.60	0.85	0.80	0.45	0.15
This thesis	0.60	0.70	0.50	0.25	0.10

Table 5.1.: Overview of the Minimum Required Overlap for $D_{\text{private}} \not\subseteq D_{\text{public}}$

Table 5.1 shows an overview of the MRO for the different datasets used and $D_{\text{private}} \not\subseteq D_{\text{public}}$ for the experiment in [SA24] compared to the base experiment of this thesis, Fig. 5.1 provides detailed success rates depending on the overlaps from 5% to 100% for all datasets used in our test run with equal salts.

5.1.1. Ground Truth Selection

To examine the effectiveness of ground truth selection, only pairs of embeddings in emb_{private} and emb_{public} are taken as input for the Wasserstein Procrustes analysis whose respective encodings match exactly. An analysis of the LSRs of ground truth selection compared to the base case scenario shows a slight, but consistent improvement. However, even for higher overlaps or larger datasets that both lead to an increased number of embeddings in the known ground truth, the MRO can only be enhanced slightly by five percentage points for every dataset evaluated. Table 5.2 summarizes this observation of slightly increased MROs, and Fig. 5.2 provides detailed insights into the different LSRs depending on the overlap of equal records. As an additional observation, note that unlike the base case, in ground truth selection, reaching the MRO is equivalent to consistently reaching LSRs of almost 100% for all overlaps $o \geq$ MRO. Thus, when applying ground truth knowledge, the matching success consistently stays at an almost perfect rate once it reaches the critical overlap.

Datasets	1000	2000	5000	10,000
Base Case	0.70	0.50	0.25	0.10
Ground Truth	0.65	0.45	0.20	0.05

Table 5.2.: Overview of the Minimum Required Overlap for Ground Truth Selection, and $D_{\text{private}} \not\subseteq D_{\text{public}}$

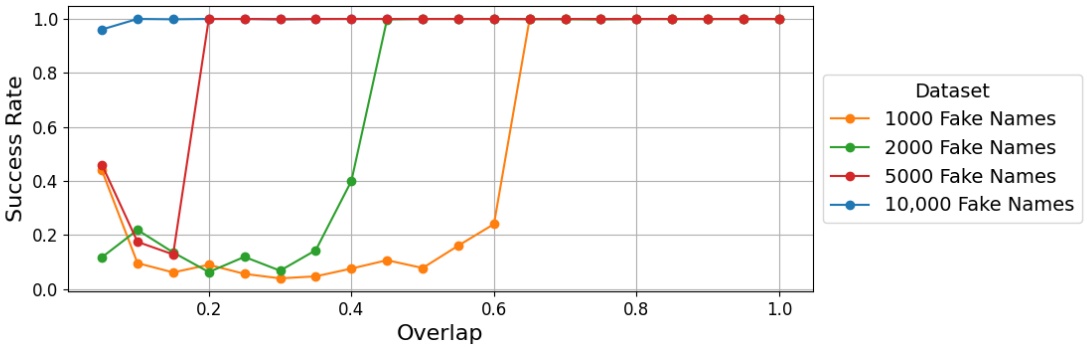

Figure 5.2.: Success Rates for Ground Truth Selection, and $D_{\text{private}} \nsubseteq D_{\text{public}}$.

Datasets	1000	2000	5000	10,000
No Dummy Values	0.65	0.45	0.20	0.05
10 Dummy Values	0.60	0.45	0.20	0.05
100 Dummy Values	0.55	0.45	0.15	0.05
1000 Dummy Values	0.05	0.05	0.05	0.05
2000 Dummy Values	0.05	0.05	0.05	0.05
5000 Dummy Values	0.05	0.05	0.05	0.05

Table 5.3.: Overview of the Minimum Required Overlap for Ground Truth Selection with Added Dummy Values, and $D_{\text{private}} \nsubseteq D_{\text{public}}$

5.1.1.1. Added Dummy Dataset for Better Ground Truth Performance

The assumption with added dummy values to the known ground truth was that it can increase the LSR as they provide additional data for the Wasserstein Procrustes analysis. This assumption turns out to be correct and is met exactly in the experimental results. Ten additional dummy values only slightly increase the MRO of the 1000 fake names dataset by five percentage points, and also 100 added dummy values are only able to improve the MRO of the 1000 and 5000 fake names datasets by 5 percentage points each, see Table 5.3. However, 1000, 2000 and 5000 dummy values all minimize the MRO to only 5%. The respective graphs in Fig. 5.3 show very clearly that this many added dummy values increase the LSR to almost 100% consistently for all datasets and overlaps.

Note that even for an overlap of only 5% on the 1000 fake names, i.e., for the smallest evaluated overlap and dataset size, the LSR is close to 100%. This is especially remarkable since these small overlaps on small datasets have performed very poorly in terms of linkage success for all other experiments conducted.

Lastly, these experiments show the strict relationship between additional dummy values and increased linkage success. In particular, dummy values do not worsen the LSR at all. This might also have been a possibility if the Wasserstein Procrustes analysis is influenced by the dummy values in a way that the embedding alignment of the original records becomes more unstable or inaccurate. However, this has been proven to not apply for GMA_{ML} and our experimental setup at all. Instead, dummy values always increase the LSR, and the more dummy values are added, the stronger the improvement.

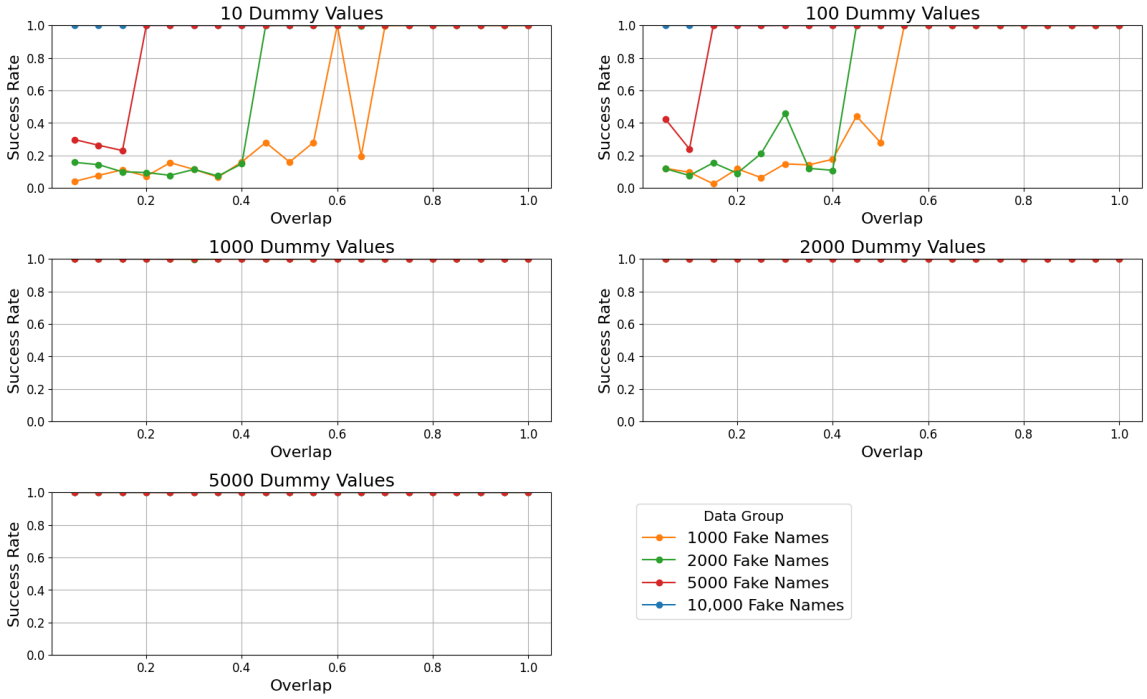

Figure 5.3.: Success Rates for Ground Truth Selection with Added Dummy Values, and $D_{\text{private}} \nsubseteq D_{\text{public}}$.

5.1.2. Comparison of Bipartite Graph Matching Techniques

An assessment of the LSRs achieved by the different bipartite graph matching techniques shows a clear outcome: The two approaches creating a full mapping, i.e., minimum weight and stable marriage matching, perform almost equally well, while symmetric and NN matching fall back significantly. Especially for small dataset sizes of up to 1000 records, symmetric and NN matching require very high overlaps of up to 100% to reach a LSR of 50% or more.

The MRO for the datasets of 2000, 5000 and 10,000 record entries are mostly comparably high for all four matching techniques, see Table 5.4. However, a closer look at Fig. 5.4 reveals a weakness in the measure of MROs: Although these minimum overlaps are reached reliably for all matching techniques and various dataset sizes, one can not assume anymore that all overlaps $o \in (0, 1], o \geq$ MRO automatically lead to LSRs $\geq 50\%$ Instead, the success rates are very unstable for symmetric and NN matching, and significant drops in linkage success can be observed, even for overlaps greater than the MRO. This shows that symmetric and NN matching are not suitable as the bipartite graph matching technique at choice due to their unpredictability, and because their MRO is higher for small datasets, and roughly equal for larger ones with at least 2000 entries. Minimum weight as well as stable marriage matching however provide comparatively equal results with similar MROs, and they are both significantly more robust.

The second characteristic to be evaluated is the FPR. An overview over the FPRs for the different matching techniques can be seen in Fig. A.2 in the appendix A.1.2. Here, one can see that the FPRs for minimum weight, stable and NN matching only drop linearly at best for increasing overlaps. For the two large datasets of 5000 and 10,000 records, symmetric

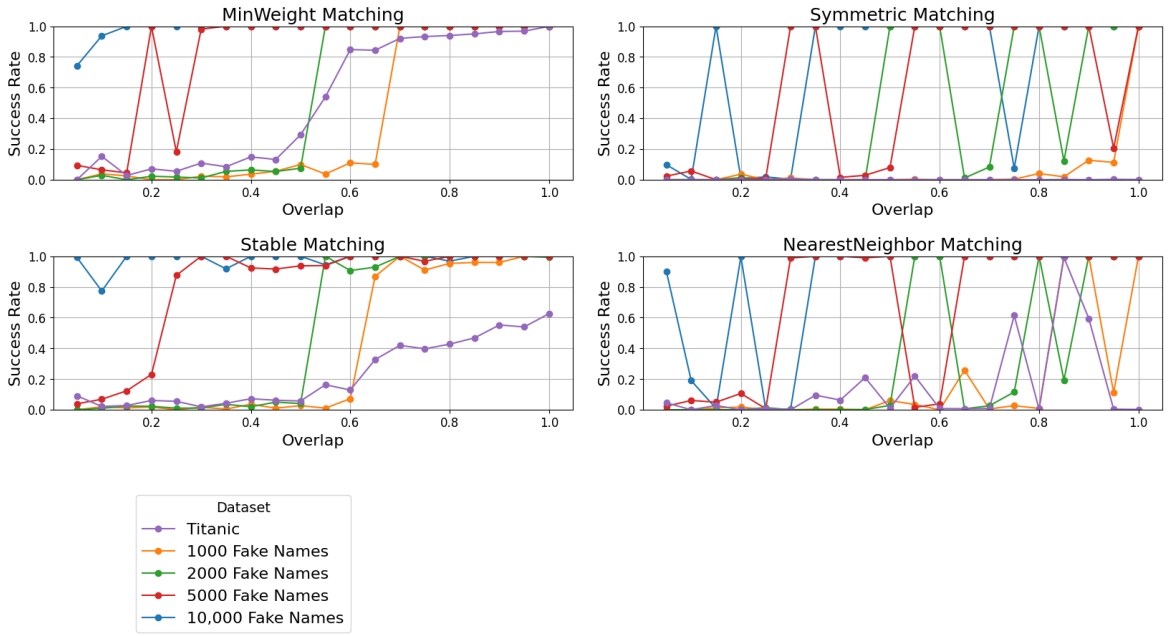

Figure 5.4.: Success Rates for Different Bipartite Graph Matching Techniques, and $D_{\text{private}} \nsubseteq D_{\text{public}}$.

matching is the only technique where the FPR drops faster than linear showing a sublinear rate of decrease, hence providing a better behaviour in terms of this measurement than any other considered graph matching algorithm. Smaller datasets though require higher overlap values for their respective FPR to drop to that level of their larger equivalents in all cases. For large enough datasets, this enables an estimation of the FPR in real-world applications. Unlike in our case where we can keep track of amongst others the FPR, it is unknown in realistic scenarios. With the clear linear and sublinear decrease rates though, one can estimate the number of false positives for the specific graph matching technique used and an estimator for the overlap rate of the two involved datasets. Finally, NN notably creates the most unstable FPR where even for high overlaps, the respective FPR jumps up irregularly to rates of up to 80% or higher.

Taking the LSR and FPR into account, minimum weight and stable bipartite graph matching are still the techniques of choice for the final matching step due to their outstanding LSRs and success stability. When dealing with datasets of at least 2000 records, and a low FPR is crucial to a specific use case, symmetric matching could be considered as an option as well.

5.1.3. Runtime Behaviour for Increasingly Larger Datasets

A runtime analysis for differently sized datasets reveals the rates by which the required time GMA_{ML} needs, increases based on the number of records that need to be linked. The asymptotic effort analysis in [SA24] showed that GMA_{ML} is of cubic time complexity as the final graph matching step is in $O(n^3)$. Fig. 5.5 shows that such a growth in time complexity cannot be observed when applying GMA_{ML} on datasets of various sizes. Rather, the rate of increase

Datasets	Titanic	1000	2000	5000	10,000
MinWeight	0.55	0.70	0.55	0.20	0.05
Stable	0.90	0.65	0.55	0.25	0.05
Symmetric	NA	1.0	0.50	0.30	0.15
Nearest Neighbour	0.75	0.85	0.55	0.30	0.05

Table 5.4.: Overview of the Minimum Required Overlap for Different Graph Matching Techniques, and $D_{\text{private}} \not\subseteq D_{\text{public}}$

in time elapsed seems to be linear, or at most of slightly higher complexity than that.

Thus, a more in-depth analysis of the main steps of GMA_{ML} can identify which steps are most crucial when it comes to the execution time of the algorithm. Fig. A.3, Fig. A.4, Fig. A.5 and Fig. A.6 in the appendix A.1.3 provide insights on the elapsed time for the four steps of GMA_{ML}: Encoding using BFs and similarity graph creation, node embedding using Node2Vec, embedding alignment using Wasserstein Procrustes analysis, and graph matching using minimum weight bipartite graph matching.

For the first step, one can very clearly see the quadratic time complexity that has already been predicted in [SA24]. Nevertheless, the required time for this step represents less than 15% of the total elapsed time and thus, even for the largest dataset of 50,000 records, its effort in computational time is by far not the driving force of the algorithm as a whole. The node embedding through Node2Vec however seems to be the most significant part when it comes to time expenditure. Almost $\frac{2}{3}$ of the total time can be traced back to this step.

Alignment through Wasserstein Procrustes analysis reveals an odd graph displaying the elapsed time in this step. On average, the aligning process does not take more than 15% of the total time. Notably though, there seem to be some outliers of irregularly high computational costs for arbitrary overlaps. They are so significant that they can also be observed in the overview of total elapsed time in Fig. 5.5 as blips in the chart. This is likely due to very unfavourable start values for the Wasserstein Procrustes analysis which results in the maximum number of epochs and iterations being executed without prior termination. Additionally, the first runs with low overlaps of 5% and 10% take very long, especially for large datasets. This is probably also due to very few overlapping values that would provide the possibility for a fast approach to good transformation matrices within the first epochs. Instead, the unsupervised Procrustes analysis needs many epochs to slowly approach better and better solutions to the Procrustes problem. This ultimately leads to significantly higher runtimes that only decrease for overlaps of 10% to 15%.

The last step of bipartite graph matching has the highest asymptotic time complexity of $O(n^3)$. In Fig. A.6 however, it becomes clear that this does not present a problem since the graph matching step only accounts for less than 10% of the total time. This step is also subject to some seemingly random fluctuations. As the time it takes is almost negligible in comparison to, most noticeable, the node embedding, these variations do not have a significant impact on the overall runtime.

5.1.4. Influence of Erroneous Datasets on the Linkage Success Rate

The last experiment has been conducted to evaluate how sensitive GMA_{ML} is to spelling errors and different naming conventions across the two datasets. As expected, errors lead to

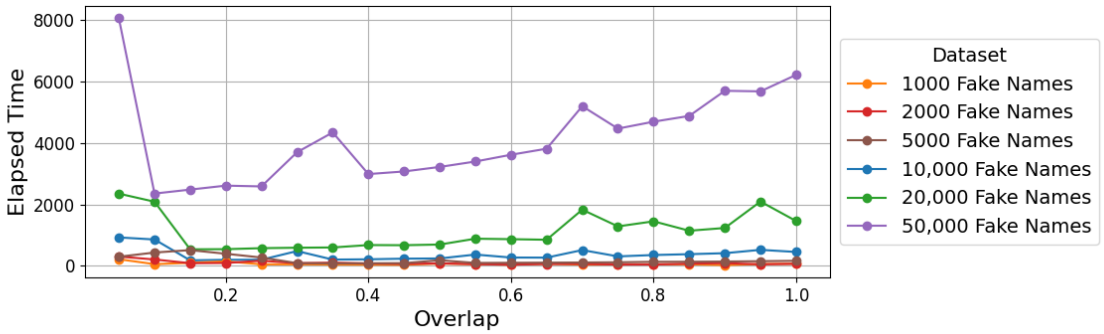

Figure 5.5.: Elapsed Time in Seconds for the Complete Execution of GMA_{ML} on Different Dataset Sizes and Overlaps.

decreased LSRs and increased MROs. Fig. 5.6 provides an overview over the LSRs for the different dataset sizes and error rates from 2% to 20%, and Table 5.5 lists the MROs.

The insights on the MROs show that errors consistently lead to higher required overlaps, despite the error rate. An exception in the collected data is the dataset of 10,000 fake names with a 2% error rate. Here, the MRO could be reduces by five percentage points from 10% to 5%. However, this is likely an outlier value and could be due to a very favourable selection of start values in the Wasserstein Procrustes analysis. Apart from that, all erroneous datasets worsen linkage performance, and the higher the error rate, the higher the MROs for each base dataset of 1000, 2000, 5000 and 10,000 records.

Additionally, one can see that for the datasets of 1000 and 2000 records, there are error rates for which the MRO is not reached at all, remarkably not even for an overlap of 100%. In the 1000 fake names dataset and given an overlap of 100% between the original and the erroneous dataset, a 15% error rate only leads to a LSR of 12%, and a 20% error rate to a LSR of only 1%. In the 2000 fake names dataset, the 15% error rate run manages to reach a MRO of 85%. However, for an error rate of 20% and for an overlap of 100%, the LSR is still only 1%.

This changes for the larger datasets of 5000 and 10,000 fake names. Although the LSRs stay significantly behind the pendants of the base case run, the MRO is reached at some point. This shows that for enough data and respectively higher overlaps, errors in record entries can be dealt with, and high LSRs can still be achieved. However, the weakness of the MRO as a success metric that has already been observed for the different graph matching techniques in subsection 5.1.2 can be seen in this experiment as well: Once again, the achieved success rates are very unstable for increasingly larger datasets and error rates, and significant drops in linkage success can be observed, even for overlaps greater than the MRO.

This could be due to an increased difficulty in the Wasserstein Procrustes analysis as it takes longer to find an acceptable transformation matrix to align D_{private} into the space of D_{public}. For good start values of the unsupervised ML Procrustes analysis, a good matrix can still be found. However, for increasingly larger datasets and error rates, the specified maximum number of epochs and iterations within the analysis likely leads to increasingly more cases where the alignment step does not compute a sufficient transformation matrix, resulting in very high fluctuations of LSRs as seen in Fig. 5.6.

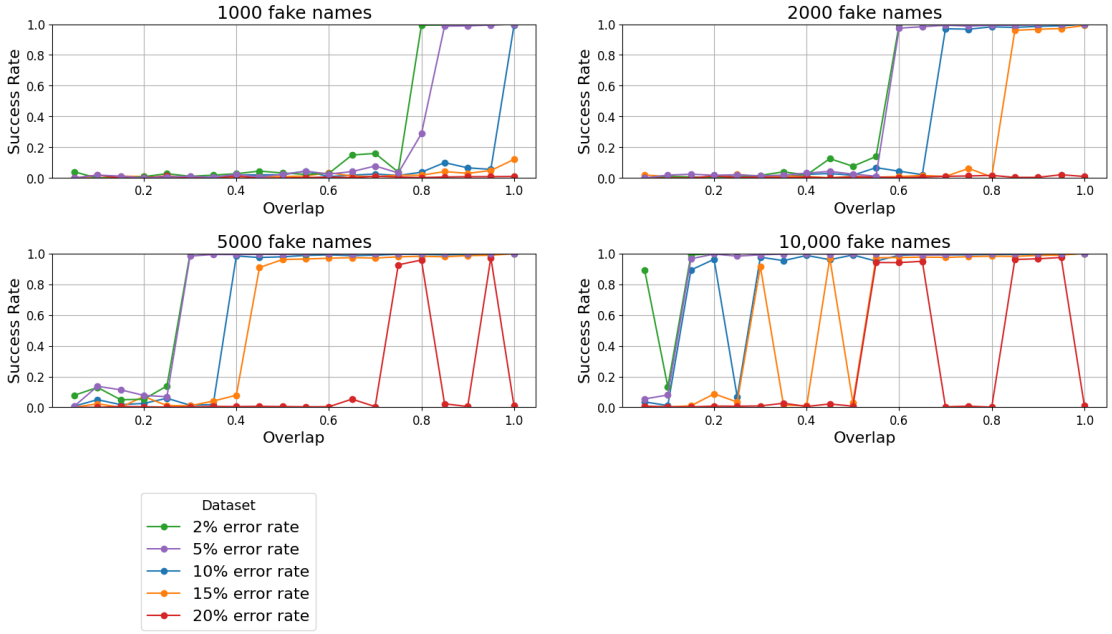

Figure 5.6.: Success Rates for Differently Erroneous Datasets, and $D_{\text{private}} \not\subseteq D_{\text{public}}$.

Datasets	1000	2000	5000	10,000
No Errors	0.70	0.50	0.25	0.10
2% Errors	0.80	0.60	0.30	0.05
5% Errors	0.85	0.60	0.30	0.15
10% Errors	1.0	0.70	0.40	0.15
15% Errors	NA	0.85	0.45	0.30
20% Errors	NA	NA	0.75	0.55

Table 5.5.: Overview of the Minimum Required Overlap for Differently Erroneous Datasets, and $D_{\text{private}} \not\subseteq D_{\text{public}}$

5.2. Discussion of the Results

The goal of this thesis was to gain a better understanding of some key properties of GMA_{ML} and to examine some possible enhancements as well as its applicability in real-world scenarios. This has been achieved fully through the successful implementation and evaluation of the four experiments of this thesis:

First, we have seen that the usage of a known ground truth as is can only slightly increase the linkage success. When adding a set number of dummy values to both datasets prior to the Wasserstein Procrustes analysis though, the linkage success can be increased: A slight improvement can be observed for 100 additional dummy values, while 1000 or more of those consistently lead to almost perfect LSRs of 100% for any overlap.

Secondly, minimum weight and stable marriage bipartite graph matching stand out as the most reliable techniques in terms of success rate and should be considered for most use cases. Symmetric matching however provides the fastest decline in the FPR.

Third, the runtime analysis revealed that node embedding through Node2Vec takes the most time out of all steps of GMA_{ML} with almost $\frac{2}{3}$ of the total runtime being traced back to this step. The most unpredictable is the Wasserstein Procrustes analysis that is mostly responsible for up to 15% of the total runtime. For very unfavourable start values of the unsupervised Procrustes analysis, this step takes significantly longer though, which can also be observed in a notable increase of the total runtime.

Lastly, increasingly erroneous records in one of the two datasets lead to worse and more unstable LSRs. For smaller datasets of up to 2000 record entries, the overlap required to reach a certain LSR increases moderately. For larger datasets of 5000 and 10,000 entries however, the results additionally become more unstable, with seemingly random drops of the LSR, even for overlaps higher than the MRO.

The results of the dummy values with known ground truth and the ones for erroneous data have been expected or predicted, and were able to be confirmed. For the usage of a known ground truth without additional dummy values, the comparison of four bipartite graph matching techniques and the runtime analysis, the experiments were more open-ended. However, their results could all be well explained and fit well into the knowledge of GMAs in general, and GMA_{ML} in particular.

The most impressing results provided the influence of additional dummy values on the LSR, as a sufficient number of added reference values lead to an almost perfect matching success, even for very small overlaps. This is particularly noteworthy as for small overlaps and relatively small datasets, GMA_{ML} is almost useless due to its very low LSR in its original form.

For the experiment on erroneous data, it should be mentioned that error rates of up to 20% are very high and can most likely be avoided in most real-world scenarios by proper data pre-processing. Consequently, more realistic error rates of up to 5%, in rare cases 10%, still lower the LSR for a given overlap, but only slightly, making an application of GMA_{ML} still useful.

6. Conclusion

This thesis provided an in-depth analysis of certain parts and behaviours of the unsupervised machine learning GMA by [SA24]. For that, the theoretical background of PPRL and GMAs as well as of the four major steps in GMA_{ML} has been explored: record encoding, node embedding, embedding alignment, and bipartite graph matching. A brief summary of related work showed that although the concept of GMAs pose a big threat to PPRL, previously proposed GMAs likely do not come to use as they require unrealistically high attacker knowledge. [SA24] though were able to outperform previous GMAs while at the same time requiring a very limited attacker model.

Afterwards, the four experiments conducted for this thesis were explained in detail, including the abstract concepts, specific implementations and considerations as well as key measurements to evaluate the success of the experiments. This evaluation followed shortly after, presenting the results in form of raw data and providing context by discussing and explaining them. As a whole, this work constitutes an exhaustive presentation, analysis and discussion of the most powerful GMA to date, providing a solid basis for further research on this or comparable GMAs, as well as applications on or countermeasures in the sense of PPRL.

6.1. Future Work

Further research based on this work could evolve around the following open questions:

This thesis and [SA24] have only considered the Wasserstein Procrustes analysis for embedding alignment. This has been proven to lead to a few, but very significant drops in the linkage success seemingly arbitrarily. Additionally, the computational costs were sometimes very high, too. While such an unsupervised, iterative Procrustes analysis is necessary when no ground truth is known, this could be avoided in exactly this case: if a common ground truth is known, a more simple orthogonal Procrustes analysis can be applied. Future experiments can show how well this kind of Procrustes analysis works to proper align the node embeddings of the two datasets, and if it can result in a significant improvement or stabilisation of the necessary runtime or resulting LSRs. In the context of Procrustes analysis, it should also be investigated in more detail how to choose the parameters of this analysis in order to find a suitable trade-off between consistently good results and computational feasibility. Thus, the arbitrarily occurring drops of the LSR could be minimized or essentially avoided.

In the domain of subsequently added dummy values, for the experiment conducted in this thesis, a static amount of 10, 100, 1000, 2000 and 5000 dummy values respectively has been chosen. This however makes it difficult for dynamic scenarios of, amongst others, different dataset sizes or differently erroneous data, to reasonably decide on the dummy dataset size for an optimal trade-off between an increased LSR and higher computational costs. Therefore, a measurement can be created, or proper estimations can be proposed as to how many dummy values are ideally required.

Our evaluation on different bipartite graph matching techniques revealed that for those techniques providing a high LSR, the FPR only decreases linearly, thus leaving especially

scenarios with low overlaps with an unsatisfactorily high FPR. Future research must investigate how to further decrease this FPR for GMA_{ML} to be able to reliably provide a certain RL quality.

Lastly, this thesis evaluated a first experiment on erroneous data. Generally speaking, this unavoidable fact that most databases contain errors, has been given too little attention in research on GMAs so far. Thus, specifically based on the experiment in this paper, the influence of both involved datasets to be erroneous on the LSR should be investigated. Additionally, for this experiment, only single characters have been changed randomly. The impact of, e.g., empty record entries, completely different naming or date conventions or data type representations on the linkage success can be taken into further consideration.

Bibliography

[AHS23] Frederik Armknecht, Youzhe Heng, and Rainer Schnell. "Strengthening Privacy-Preserving Record Linkage using Diffusion." en. In: *Proceedings on Privacy Enhancing Technologies* 2023.2 (Apr. 2023), pp. 298–311. ISSN: 2299-0984. DOI: 10.56553/popets-2023-0054. URL: https://petsymposium.org/popets/2023/popets-2023-0054.php (visited on 06/26/2024).

[AP21] Ridho Ananda and Agi Prasetiadi. "Classification based on configuration objects by using Procrustes analysis." In: *Jurnal Infotel* 13.2 (2021), pp. 76–83.

[Bro97] A.Z. Broder. "On the resemblance and containment of documents." In: *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No.97TB100171)*. 1997, pp. 21–29. DOI: 10.1109/SEQUEN.1997.666900.

[BV10] J. Bijsterbosch and A. Volgenant. "Solving the Rectangular assignment problem and applications." In: *Annals of Operations Research* 181.1 (Dec. 2010), pp. 443–462. ISSN: 1572-9338. DOI: 10.1007/s10479-010-0757-3. URL: https://doi.org/10.1007/s10479-010-0757-3.

[Chr+18] Peter Christen et al. "Pattern-Mining Based Cryptanalysis of Bloom Filters for Privacy-Preserving Record Linkage." In: *Advances in Knowledge Discovery and Data Mining*. Ed. by Dinh Phung et al. Cham: Springer International Publishing, 2018, pp. 530–542. ISBN: 978-3-319-93040-4.

[Cro16] David F. Crouse. "On implementing 2D rectangular assignment algorithms." In: *IEEE Transactions on Aerospace and Electronic Systems* 52.4 (2016), pp. 1679–1696. DOI: 10.1109/TAES.2016.140952.

[CRS22] Peter Christen, Thilina Ranbaduge, and Rainer Schnell. "Linking Sensitive Data: Methods and Techniques for Practical Privacy-Preserving Information Sharing: Synopsis by Kerina Jones." In: *International Journal of Population Data Science* 6.2 (June 2022). URL: https://ijpds.org/article/view/1657.

[CRT17] Chris Culnane, Benjamin I. P. Rubinstein, and Vanessa Teague. *Vulnerabilities in the use of similarity tables in combination with pseudonymisation to preserve data privacy in the UK Office for National Statistics' Privacy-Preserving Record Linkage*. en. arXiv:1712.00871 [cs]. Dec. 2017. URL: http://arxiv.org/abs/1712.00871 (visited on 06/26/2024).

[Cur18] Dylan Curran. *Are you ready? Here is all the data Facebook and Google have on you*. Accessed: 2024-07-04. 2018. URL: https://www.theguardian.com/commentisfree/2018/mar/28/all-the-data-facebook-google-has-on-you-privacy.

[DM04a] Peter C. Dillinger and Panagiotis Manolios. "Bloom Filters in Probabilistic Verification." In: *Formal Methods in Computer-Aided Design*. Ed. by Alan J. Hu and Andrew K. Martin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 367–381. ISBN: 978-3-540-30494-4.

[DM04b] Peter C. Dillinger and Panagiotis Manolios. "Fast and Accurate Bitstate Verification for SPIN." In: *Model Checking Software*. Ed. by Susanne Graf and Laurent Mounier. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 57–75. ISBN: 978-3-540-24732-6.

[Dun46] Halbert L Dunn. "Record linkage." In: *American Journal of Public Health and the Nations Health* 36.12 (1946), pp. 1412–1416.

[EPY97] D. Eppstein, M. S. Paterson, and F. F. Yao. "On Nearest-Neighbor Graphs." In: *Discrete & Computational Geometry* 17.3 (Apr. 1997), pp. 263–282. ISSN: 1432-0444. DOI: 10.1007/PL00009293. URL: https://doi.org/10.1007/PL00009293.

[Est17] Asunción Esteve. "The business of personal data: Google, Facebook, and privacy issues in the EU and the USA." In: *International Data Privacy Law* 7.1 (2017), pp. 36–47.

[FKP12] Michael Fleming, Brad Kirby, and Kay I Penny. "Record linkage in Scotland and its applications to health research." In: *Journal of clinical nursing* 21.19pt20 (2012), pp. 2711–2721.

[GB06] Lifang Gu and Rohan Baxter. "Decision Models for Record Linkage." In: *Data Mining: Theory, Methodology, Techniques, and Applications*. Ed. by Graham J. Williams and Simeon J. Simoff. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 146–160. ISBN: 978-3-540-32548-2. DOI: 10.1007/11677437_12. URL: https://doi.org/10.1007/11677437_12.

[GD+21] Aris Gkoulalas-Divanis et al. "Modern Privacy-Preserving Record Linkage Techniques: An Overview." en. In: *IEEE Transactions on Information Forensics and Security* 16 (2021), pp. 4966–4987. ISSN: 1556-6013, 1556-6021. DOI: 10.1109/TIFS.2021.3114026. URL: https://ieeexplore.ieee.org/document/9541149/ (visited on 06/26/2024).

[GJB19] Edouard Grave, Armand Joulin, and Quentin Berthet. "Unsupervised Alignment of Embeddings with Wasserstein Procrustes." In: *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*. Ed. by Kamalika Chaudhuri and Masashi Sugiyama. Vol. 89. Proceedings of Machine Learning Research. PMLR, Apr. 2019, pp. 1880–1890. URL: https://proceedings.mlr.press/v89/grave19a.html.

[GL16] Aditya Grover and Jure Leskovec. *node2vec: Scalable Feature Learning for Networks*. en. arXiv:1607.00653 [cs, stat]. July 2016. URL: http://arxiv.org/abs/1607.00653 (visited on 06/26/2024).

[Gu+03] Lifang Gu et al. "Record linkage: Current practice and future directions." In: *CSIRO Mathematical and Information Sciences Technical Report* 3 (2003), p. 83.

[Hen+22] Youzhe Heng et al. "On the effectiveness of graph matching attacks against privacy-preserving record linkage." en. In: *PLOS ONE* 17.9 (Sept. 2022). Ed. by Muhammad Khurram Khan, e0267893. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0267893. URL: https://dx.plos.org/10.1371/journal.pone.0267893 (visited on 06/26/2024).

[Lev+66] Vladimir I Levenshtein et al. "Binary codes capable of correcting deletions, insertions, and reversals." In: *Soviet physics doklady*. Vol. 10. 8. Soviet Union. 1966, pp. 707–710.

[LH13] Baoli Li and Liping Han. "Distance Weighted Cosine Similarity Measure for Text Classification." In: *Intelligent Data Engineering and Automated Learning – IDEAL 2013*. Ed. by Hujun Yin et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 611–618. ISBN: 978-3-642-41278-3.

[Mik+13] Tomas Mikolov et al. *Efficient Estimation of Word Representations in Vector Space*. en. arXiv:1301.3781 [cs]. Sept. 2013. URL: http://arxiv.org/abs/1301.3781 (visited on 05/02/2024).

[Mit+17] William Mitchell et al. "A graph traversal attack on Bloom filter-based medical data aggregation." In: *Int. J. Big Data Intell.* 4 (2017), pp. 217–226. URL: https://api.semanticscholar.org/CorpusID:9682742.

[Pal+18] Enrico Palumbo et al. "Knowledge Graph Embeddings with node2vec for Item Recommendation." In: *The Semantic Web: ESWC 2018 Satellite Events*. Ed. by Aldo Gangemi et al. Cham: Springer International Publishing, 2018, pp. 117–120. ISBN: 978-3-319-98192-5.

[PGS19] Jiajie Peng, Jiaojiao Guan, and Xuequn Shang. "Predicting Parkinson's Disease Genes Based on Node2vec and Autoencoder." In: *Frontiers in Genetics* 10 (2019). ISSN: 1664-8021. DOI: 10.3389/fgene.2019.00226. URL: https://www.frontiersin.org/journals/genetics/articles/10.3389/fgene.2019.00226.

[Pit92] Boris Pittel. "On Likely Solutions of a Stable Marriage Problem." In: *The Annals of Applied Probability* 2.2 (1992), pp. 358 –401. DOI: 10.1214/aoap/1177005708. URL: https://doi.org/10.1214/aoap/1177005708.

[RCS20] Thilina Ranbaduge, Peter Christen, and Rainer Schnell. "Secure and Accurate Two-Step Hash Encoding for Privacy-Preserving Record Linkage." In: *Advances in Knowledge Discovery and Data Mining*. Ed. by Hady W. Lauw et al. Cham: Springer International Publishing, 2020, pp. 139–151. ISBN: 978-3-030-47436-2.

[RTG98] Y. Rubner, C. Tomasi, and L.J. Guibas. "A metric for distributions with applications to image databases." In: *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*. 1998, pp. 59–66. DOI: 10.1109/ICCV.1998.710701.

[SA24] Jochen Schäfer and Frederik Armknecht. "Revisiting Graph Matching Attacks Against Non-Interactive Privacy-Preserving Record Linkage." en. In: *tbd* (2024).

[SB16] Rainer Schnell and Christian Borgs. "Randomized Response and Balanced Bloom Filters for Privacy Preserving Record Linkage." In: *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. 2016, pp. 218–224. DOI: 10.1109/ICDMW.2016.0038.

[Smi17] D. Smith. "Secure pseudonymisation for privacy-preserving probabilistic record linkage." In: *Journal of Information Security and Applications* 34 (2017), pp. 271–279. ISSN: 2214-2126. DOI: https://doi.org/10.1016/j.jisa.2017.01.002. URL: https://www.sciencedirect.com/science/article/pii/S2214212616301405.

[Tho17] Mikkel Thorup. "Fast and powerful hashing using tabulation." In: *Commun. ACM* 60.7 (June 2017), 94–101. ISSN: 0001-0782. DOI: 10.1145/3068772. URL: https://doi.org/10.1145/3068772.

[TIR78] Steven L. Tanimoto, Alon Itai, and Michael Rodeh. "Some Matching Problems for Bipartite Graphs." In: *J. ACM* 25.4 (Oct. 1978), 517–525. ISSN: 0004-5411. DOI: 10.1145/322092.322093. URL: https://doi.org/10.1145/322092.322093.

[TJ13] Vikas Thada and Vivek Jaglan. "Comparison of jaccard, dice, cosine similarity coefficient to find best fitness value for web retrieved documents using genetic algorithm." In: *International Journal of Innovations in Engineering and Technology* 2.4 (2013), pp. 202–205.

[VCV13] Dinusha Vatsalan, Peter Christen, and Vassilios S. Verykios. "A taxonomy of privacy-preserving record linkage techniques." en. In: *Information Systems* 38.6 (Sept. 2013), pp. 946–969. ISSN: 03064379. DOI: 10.1016/j.is.2012.11.005. URL: https://linkinghub.elsevier.com/retrieve/pii/S0306437912001470 (visited on 06/26/2024).

[Vid+19] Anushka Vidanage et al. "Efficient Pattern Mining Based Cryptanalysis for Privacy-Preserving Record Linkage." In: *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. 2019, pp. 1698–1701. DOI: 10.1109/ICDE.2019.00176.

[Vid+20] Anushka Vidanage et al. "A Graph Matching Attack on Privacy-Preserving Record Linkage." en. In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. Virtual Event Ireland: ACM, Oct. 2020, pp. 1485–1494. ISBN: 978-1-4503-6859-9. DOI: 10.1145/3340531.3411931. URL: https://dl.acm.org/doi/10.1145/3340531.3411931 (visited on 06/26/2024).

A. Appendix

A.1. More Detailed Insights on Some of the Conducted Experiments

A.1.1. Confirmation of the Results Obtained by Schäfer et al.

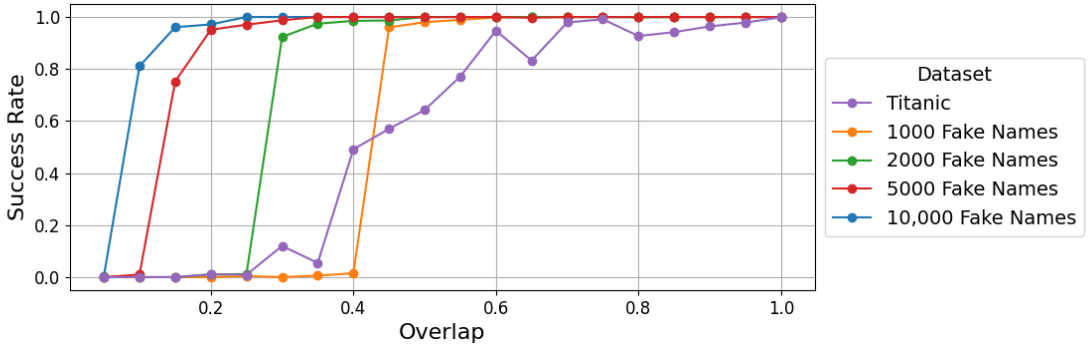

Figure A.1.: Success Rates for the Base Case Scenario, and $D_{\text{private}} \subseteq D_{\text{public}}$.

Datasets	Titanic	1000	2000	5000	10,000
[SA24]	0.50	0.45	0.35	0.15	0.15
This thesis	0.45	0.45	0.30	0.15	0.10

Table A.1.: Overview of the Minimum Required Overlap for for Base Case Scenario, and $D_{\text{private}} \subseteq D_{\text{public}}$

A.1.2. Comparison of Bipartite Graph Matching Techniques

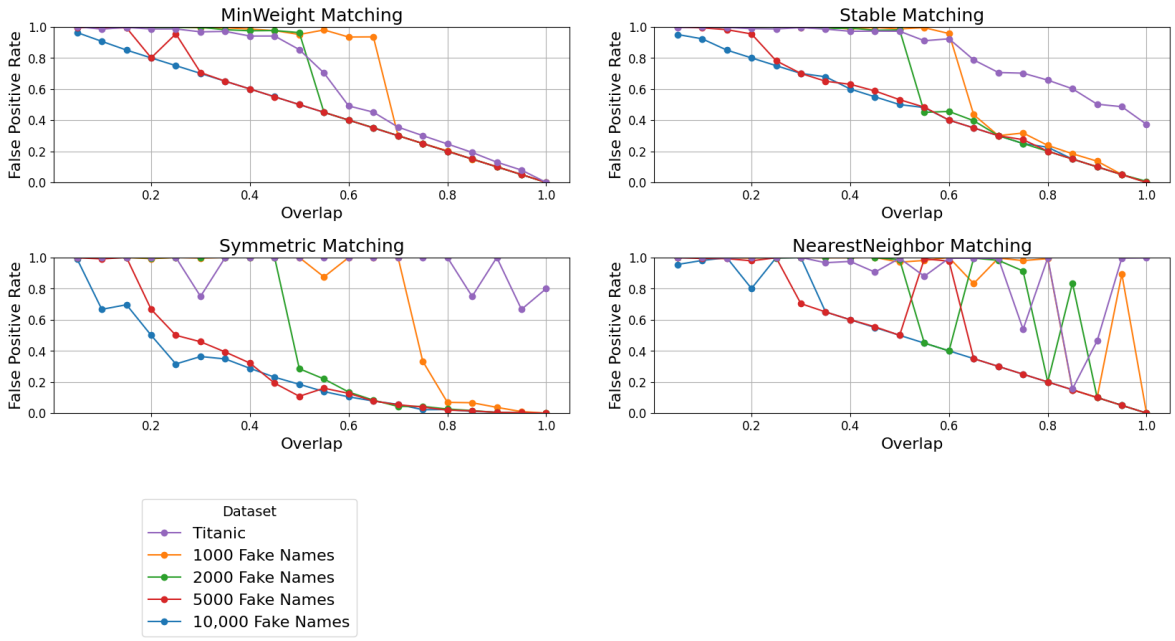

Figure A.2.: False Positive Rates for Different Bipartite Graph Matching Techniques, and $D_{\text{private}} \nsubseteq D_{\text{public}}$.

A.1.3. Runtime Analysis for Increasingly Larger Datasets

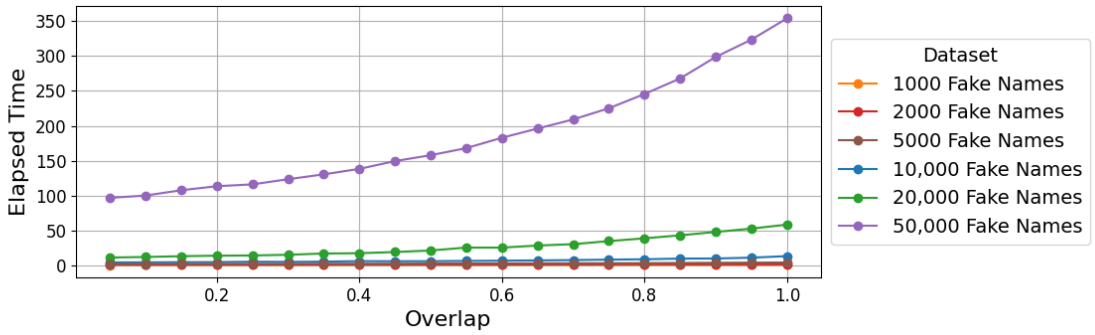

Figure A.3.: Elapsed Time in Seconds for the Encoding and Similarity Graph Creation of D_{private} on Different Dataset Sizes and Overlaps.

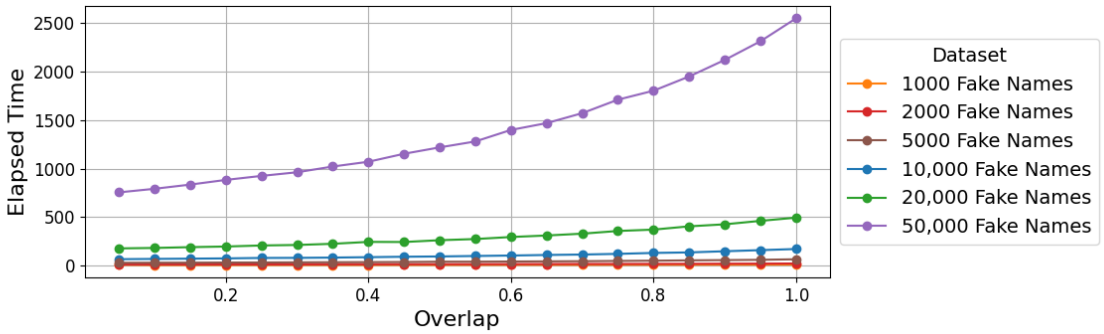

Figure A.4.: Elapsed Time in Seconds for the Embedding of D_{private} on Different Dataset Sizes and Overlaps.

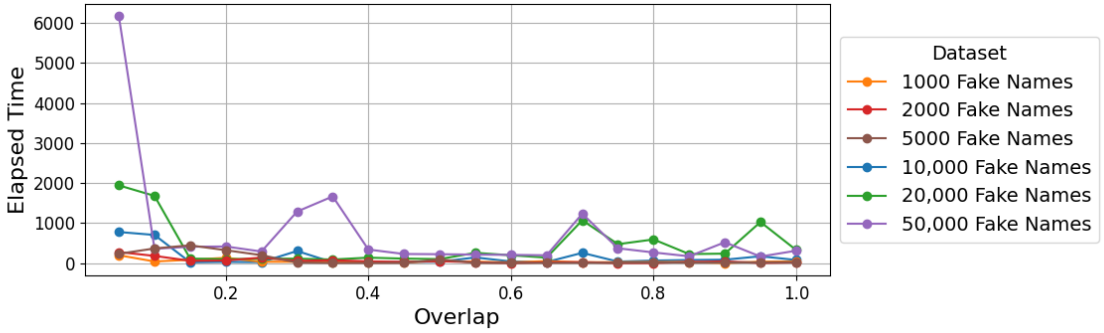

Figure A.5.: Elapsed Time in Seconds for the Embedding Alignment Step Using Wasserstein Procrustes on Different Dataset Sizes and Overlaps.

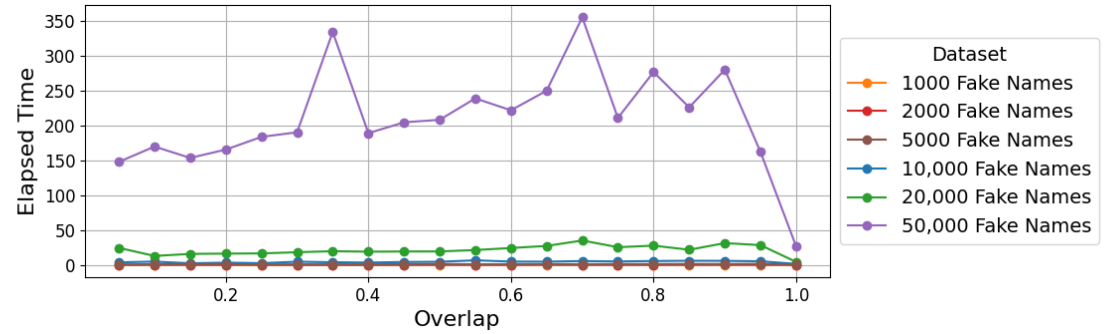

Figure A.6.: Elapsed Time in Seconds for the Final Graph Matching Step Using Minimum Weight Bipartite Graph Matching on Different Dataset Sizes and Overlaps.

Eidesstattliche Erklärung

Hiermit versichere ich, dass diese Abschlussarbeit von mir persönlich verfasst ist und dass ich keinerlei fremde Hilfe in Anspruch genommen habe. Ebenso versichere ich, dass diese Arbeit oder Teile daraus weder von mir selbst noch von anderen als Leistungsnachweise andernorts eingereicht wurden. Wörtliche oder sinngemäße Übernahmen aus anderen Schriften und Veröffentlichungen in gedruckter oder elektronischer Form sind gekennzeichnet. Sämtliche Sekundärliteratur und sonstige Quellen sind nachgewiesen und in der Bibliographie aufgeführt. Das Gleiche gilt für graphische Darstellungen und Bilder sowie für alle Internet-Quellen.
Ich bin ferner damit einverstanden, dass meine Arbeit zum Zwecke eines Plagiatsabgleichs in elektronischer Form anonymisiert versendet und gespeichert werden kann.

16. Juli 2024 — Kilian Hüllen

Datum — Kilian Hüllen