

PhD Thesis

Efficient Techniques for Privacy Preserved Incremental Record Linkage of Noisy Health Data

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirement for the degree of

DOCTOR OF PHILOSOPHY
IN
COMPUTER SCIENCE AND ENGINEERING

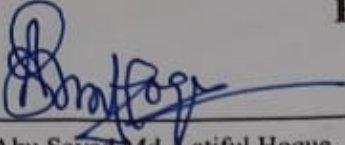
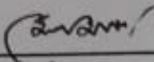
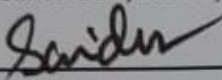
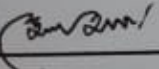
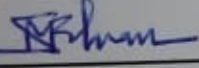
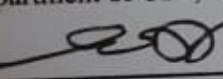
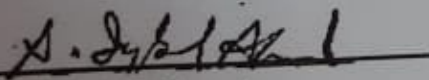
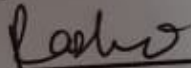
by
Shahidul Islam Khan
Student ID 0413054002F

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
BANGLADESH UNIVERSITY OF ENGINEERING AND
TECHNOLOGY
DHAKA 1000, BANGLADESH

February 2020

The thesis titled "Efficient Techniques for Privacy Preserved Incremental Record Linkage of Noisy Health Data" submitted by Shahidul Islam Khan, Roll No. 0413054002F, Session April 2013, to the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science and Engineering and approved as to its style and contents on February 29, 2020.

Board of Examiners

1. 
Dr. Abu Sayed Md. Latiful Hoque
Professor
Department of CSE, BUET, Dhaka. Chairman
(Supervisor)
2. 
Head (Ex-officio)
Department of CSE, BUET, Dhaka.
3. 
Dr. Md. Saidur Rahman Member
Professor
Department of CSE, BUET, Dhaka.
4. 
Dr. Md. Mostofa Akbar Member
Professor
Department of CSE, BUET, Dhaka.
5. 
Dr. M. Sohel Rahman Member
Professor
Department of CSE, BUET, Dhaka.
6. 
Dr. Mohammed Eunus Ali Member
Professor
Department of CSE, BUET, Dhaka.
7. 
Dr. Sheikh Iqbal Ahamed Member (External)
Professor and Chair
Department of Computer Science
Marquette University, Wisconsin, USA.
8. 
Dr. Mohammad Rashedur Rahman Member (External)
Professor
Department of Electrical and Computer Engineering
North South University, Dhaka.

Candidate's Declaration

This is to certify that the work entitled "Efficient Techniques for Privacy Preserved Incremental Record Linkage of Noisy Health Data" is the outcome of the research carried out by me under the supervision of Prof. Dr. Abu Sayed Md. Latiful Hoque in the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka-1000. It is also declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.



Shahidul Islam Khan
Candidate

Dedication

To my Parents, Wife and Children

Contents

<i>Board of Examiners</i>	i
<i>Candidate's Declaration</i>	ii
<i>Candidate's Dedication</i>	iii
Acknowledgments	xiii
Abstract	xiv
1 Introduction	1
1.1 Background and Motivation	2
1.1.1 Application Area	3
1.2 Research Problem and Methodology	4
1.3 Contributions	5
1.4 Organization of the Thesis	6
2 Background Study	7
2.1 Record Linkage	7
2.1.1 Data Preprocessing	8
2.1.2 Blocking	9
2.1.3 Comparison	11
2.1.4 Classification	12
2.1.5 Evaluation	13
2.2 Privacy Preserving Record Linkage	14
2.2.1 Privacy Techniques	16
2.2.2 Privacy Attacks	18
2.2.3 Stakeholders in PPRL	19
2.2.4 Adversary Models	20
2.3 Health Records and Health Data Server: Privacy and Security Issues	21
2.3.1 Data Breaches of Health Information Systems	23
2.3.2 Some Incidents of Health Data Breaches and Factors Behind	23
2.3.3 Cyber Attacks on Healthcare Servers	24

2.3.4	Other Impacts of Health Data Breaches	25
2.3.5	Analysis of the Risks of Health Information Systems	25
2.4	Constraints of Health Data in Developing Countries	26
2.5	Healthcare Data Generation Scenario in Bangladesh	28
2.5.1	Patient treatment cycle in Bangladesh	28
2.5.2	Record Linkage Problem	30
2.6	Summary	32
3	Missing Data Imputation	33
3.1	Introduction	33
3.2	Background and Related Works	35
3.2.1	Single Imputation	36
3.2.2	Multiple Imputation	40
3.2.2.1	Predictive Mean Matching	43
3.2.2.2	Logistic Regression	43
3.2.2.3	Polytomous Logistic Regression	43
3.2.2.4	Linear Discriminant Analysis	44
3.2.2.5	Classification and Regression Tree	44
3.2.2.6	Bayesian Linear Regression	45
3.2.2.7	Amelia	45
3.2.3	Summary of Literature Review and Research Gap	45
3.3	Proposed algorithm	45
3.4	Experimental Design	47
3.4.1	Description of the Datasets	49
3.4.1.1	Local Health Dataset	49
3.4.1.2	Hair Eye Color Dataset	50
3.4.1.3	UCI Car Dataset	50
3.4.1.4	Kaggle House Price Dataset	50
3.5	Results	50
3.5.1	Performance Comparison for Binary Attributes	51
3.5.2	Performance Comparison for Ordinal Attribute	54
3.5.3	Performance Comparison for Numeric Attribute	56
3.6	Discussions and Limitation	58
3.7	Summary	59
4	Phonetic Encoding for Record Linkage	60
4.1	Introduction	61
4.1.1	Significance of Bengali Language	62
4.1.2	A Motivational Example	62
4.1.3	Contributions	64
4.2	Literature Review	65
4.2.1	Widely Used Phonetic Algorithms	65
4.2.1.1	Soundex	66
4.2.1.2	NYSIIS	66

4.2.1.3	Match Rating Codex	66
4.2.1.4	Metaphone	67
4.2.2	Algorithms for Bengali Phonetic Names	67
4.2.2.1	NameValue and NameSig	67
4.2.2.2	Modified NameSignificance	69
4.2.2.3	Double Metaphone Encoding Technique	69
4.2.2.4	Bengali Phonetic Encoding	70
4.2.3	Summary of Literature Review and Research Gap	70
4.3	Proposed Algorithm: nameGist	70
4.3.1	Vowel Marks	71
4.3.2	Similar Sounding Alphabet Mapping	72
4.3.2.1	Similar Sounding Vowel Mapping	72
4.3.2.2	Similar Sounding Consonant Mapping	73
4.3.3	Algorithm: nameGist	73
4.4	Description of the Datasets	77
4.4.1	Bengali Name in English Representation (Bengali Phonetic) Dataset	77
4.4.2	American/British Name Dataset	77
4.4.3	Bengali Unicode Name Dataset	78
4.4.4	Bengali Phonetic, Bengali Unicode, and US/UK English Mixed Name Dataset	79
4.5	Results and Discussion	79
4.5.1	Experimental Setup	79
4.5.1.1	Working Environment	80
4.5.1.2	Algorithm Implementation Information	80
4.5.1.3	Measurements	80
4.5.2	Results	82
4.5.2.1	Bengali Name in English Representation (Bengali Phonetic)	82
4.5.2.2	English (British/American) Names	82
4.5.2.3	Bengali Name in Bengali Representation (Bengali Unicode)	85
4.5.2.4	English and Bengali Mix Dataset	85
4.5.3	Discussion	85
4.5.4	Limitation	87
4.6	Conclusion	90
5	Key-based Secured Record Linkage	91
5.1	Introduction	92
5.2	Proposed Solution: Key-based Secured Record Linkage (KSRL) Technique	94
5.2.1	Changeable Attributes	94
5.2.2	Fixed Unambiguous Attributes	95
5.2.3	Ambiguous Attributes	97
5.2.4	NameValue Generation	98
5.2.5	KSRL Key Generation	98
5.3	Results and Discussion	101
5.3.1	Clustering Analysis	103

5.3.2	Finding the Significance of the Attributes	103
5.3.3	Effects of Privacy Preservation over Record Linkage Performance	104
5.3.4	Privacy Evaluation	105
5.4	Limitation	106
5.5	Summary	106
6	Incremental Record Linkage with Privacy Preservation	108
6.1	Introduction	109
6.2	Literature Review	110
6.2.1	Record Linkage	110
6.2.2	Privacy Preserving Record Linkage	111
6.2.3	Incremental Record Linkage	111
6.2.4	Summary of Literature Review and Research Gap	112
6.3	Background Knowledge and Problem Formulation	112
6.4	PPiRL, an End-to-End Framework	116
6.4.1	Data Pre-processing	116
6.4.2	Privacy Preservation	119
6.4.2.1	Phonetic Encoding	119
6.4.2.2	K-anonymization Method	119
6.4.3	Blocking	122
6.4.3.1	Traditional Blocking	122
6.4.4	Clustering	125
6.4.5	Evaluation	127
6.4.5.1	Linkage Evaluation	127
6.4.5.2	Privacy Evaluation	127
6.5	Experimental Results	127
6.5.1	Data Pre-processing	128
6.5.1.1	Feature Selection	128
6.5.1.2	Normalization of Age Values	129
6.5.1.3	Address Standardization	130
6.5.2	Experimental Setup	131
6.5.3	Linkage Evaluation	132
6.5.3.1	External validation Measure Results	132
6.5.3.2	Internal Validation Measure Results	133
6.5.4	Privacy Evaluation	134
6.5.4.1	Frequency Analysis	134
6.5.4.2	Dictionary Attack	136
6.5.4.3	Information Gain	136
6.5.5	Comparison of PPiRL with Batch Record Linkage	138
6.5.6	Comparison of PPiRL with Incremental Record Linkage	139
6.6	Summary	139

7 Conclusions	141
7.1 Missing Data Imputation	141
7.2 Phonetic Encoding	142
7.3 Key Based Privacy Preserving Record Linkage	143
7.4 Incremental Privacy Preserving Record Linkage	143
7.5 Broader Impacts of the Thesis	144
7.6 Future Works	145
Publications	147
Bibliography	147

List of Figures

2.1	Outline of the general record linkage process	8
2.2	Outline of privacy preserved record linkage (PPRL) process adopted from [183]	15
2.3	A taxonomy of PPRL techniques adopted from [183]	16
2.4	Main causes of data breach in the healthcare industry	24
2.5	Patient treatment in a hospital as an outdoor patient	29
2.6	Patient treatment in a hospital as an indoor patient	29
2.7	Patient treatment in a diagnostic center	30
2.8	Patient treatment in a doctors private chamber	31
3.1	Regression lines from two sets of random 100 data taken from 1000 library fine data	40
3.2	Flowchart of MICE	42
3.3	Flowchart of SICE	48
3.4	Block diagram of the system	49
3.5	Accuracy and F-measure for four algorithms to impute gender attribute . . .	53
3.6	Performance comparison of MICE and SICE for additional binary datasets .	53
3.7	Performance of MICE and SICE for ordinal data using PMM and POLYREG	55
3.8	Comparison of execution time of MICE and SICE to impute UCI car dataset	56
3.9	Performance of algorithms to predict house prices	58
4.1	Vowel marks in phonetic and unicode Bengali name	71
4.2	Steps of nameGist algorithm for names written in English	75
4.3	Used steps of nameGist algorithm applicable for names written in Bengali .	76
4.4	Accuracy and F1 score of all algorithms for English name dataset 1	83
4.5	Accuracy and F1 score of all algorithms for English name dataset 2	83
5.1	Block diagram of PITSRL	93
5.2	Block diagram of key-based secured record linkage	101
5.3	Frequency analysis	105
6.1	Stages of batch record linkage	113
6.2	Stages of incremental record linkage	115
6.3	PPiRL, an end-to-end framework steps for the base dataset	117
6.4	PPiRL, an end-to-end framework steps for increments	118
6.5	Process of phonetic encoding	120

6.6	Illustration of k-anonymization process	121
6.7	Traditional blocking using Soundex code	124
6.8	Frequency analysis of original 'Name' attribute	135
6.9	Frequency analysis of 'Name' attribute after privacy enforcement	136

List of Tables

2.1	Statistics of healthcare server attack compared to total healthcare breach . . .	25
2.2	Ambiguity in patients' name imputation	27
3.1	A dataset with missing values	36
3.2	Imputing missing values using single imputation method	37
3.3	Analysis of bias for single imputation method	38
3.4	Example of 1000 library fine data with missing values	41
3.5	Multiple imputation for table 3.4	41
3.6	List of existing algorithms implemented for comparison	51
3.7	Datasets used for imputation of binary attribute	52
3.8	Results for binary dataset "gender"	52
3.9	Performance of MICE and SICE for ordinal attribute using local health dataset	54
3.10	Performance of MICE and SICE for ordinal attribute using UCI car dataset	55
3.11	Performance of the algorithms for numeric attribute of local health dataset .	57
4.1	Bengali vowels and other miscellaneous characters	63
4.2	Bengali consonants	64
4.3	Example of misspelling	65
4.4	Conversion of names with vowel marks and spelling variations to the "gist" name	72
4.5	Vowel marks mapping	72
4.6	Similar sounding vowel mapping	73
4.7	Similar sounding consonant mapping	74
4.8	Sample of Bengali phonetic dataset	77
4.9	Sample of US/UK English name dataset	78
4.10	Sample of Bengali unicode dataset	78
4.11	English- Bengali mix dataset	79
4.12	Description of the confusion matrix	80
4.13	Result of Bengali phonetic name dataset	83
4.14	Result of English name dataset 1	84
4.15	Result of English name dataset 2	84
4.16	Result of Bengali unicode dataset	85
4.17	Result of English and Bengali mix dataset	86
4.18	Similar sounding names with different spellings	88

4.19	Examples of failures	89
5.1	Gender code	95
5.2	Conversion of date of birth to birth year range	96
5.3	Age range	96
5.4	Conversion of date of birth to age range	96
5.5	Fixed unambiguous attributes	97
5.6	Illustration of significant ambiguous hashed value selection	99
5.7	Hashed address	99
5.8	Sample KSRL-key	100
5.9	Patient dataset analysis	102
5.10	Sample records for key generation	104
5.11	Impact of attributes for record linkage	104
6.1	Use of surnames as blocking keys	123
6.2	Performance of the blocking keys	125
6.3	Feature selection	129
6.4	Age normalization	130
6.5	Mapping address to geocode	131
6.6	PPiRL performance on real data with various noise setting	133
6.7	Performance of PPiRL using synthetic dataset	133
6.8	Penalty evaluation for naive and correlation clustering	134
6.9	Entropy of individual attributes and concatenated data	137
6.10	Information gain	138
6.11	Comparison between PPiRL and batch record linkage	139
6.12	Comparison between PPiRL and IRL	140

Acknowledgments

All praises due to Allah, the most benevolent and merciful.

I thank Allah ta'ala for His mercy on me always, especially during my PhD journey. Alhamdulillah for always giving me what I needed instead of what I wanted. I want to express my deepest gratitude to my supervisor, Professor Dr. Abu Sayed Md. Latiful Hoque for guiding me in this long PhD journey and showing me how to conduct successful research and, above all, for always being there as a mentor. His suggestions encouraged me to identify novel problems, analyze them deeply, and solve them properly. He taught me, step-by-step, how to write academic papers. His support gave me strength at the time of my disappointment. Without his continuous motivation and encouragement, I could not have finished this writing.

I am grateful to the members of my Doctoral Committee for their valuable feedback during my research, which improves the quality of my thesis. I thank the Department of CSE, BUET, for providing me with state-of-the-art facilities. I appreciate all the teachers of the Department for inspiring me to remain motivated. I thank the authorities of the International Islamic University Chittagong (IIUC) for providing me Study Leave to pursue my PhD. I sincerely thank my parents for their love, support, and doa for me. I thank my wife and children for their patience, sacrifices, and understanding during my long journey towards the PhD.

This research is funded under the PhD Fellowship scheme of the ICT Division, Ministry of Posts, Telecommunications and Information Technology, Government of the People's Republic of Bangladesh. I convey my heartfelt gratitude to all the persons related to this fellowship scheme.

Abstract

At present many public and private organizations collect a huge amount of data. Later, these data are processed and analyzed to discover interesting knowledge that supports proper decision making. Developing efficient techniques for cleaning and linking large datasets to support knowledge discovery has gained high importance in both academia and industry. Solving record linkage problems with an incremental approach is a relatively new research area. Few studies have been performed in the field of incremental record linkage targeting the linkage quality or efficiency. However, the privacy issue regarding the incremental approach has not yet been discussed. Privacy preservation is essential for sensitive record linkage, e.g., health records, financial records, etc. In this regard, we have come up with a novel concept which encapsulates privacy-preserving techniques with an incremental record linkage approach.

In this thesis, we focus on the healthcare domain. A problem with real health data is that it is noisy by nature. Another problem with health data is the presence of missing values. We propose a novel phonetic algorithm to reduce the noise in patients' names to improve the performance of record linkage. For handling missing data, we extend the widely used MICE algorithm to impute missing data of both categorical and numeric attributes.

For preserving privacy, we use different privacy techniques such as phonetic encoding, hashing, and generalization. For handling incremental updates and internal linkage, we use the Naive incremental clustering approach. We perform various experiments to test the privacy and linkage quality of our proposed framework. We compare our work with the existing incremental record linkage framework and also with existing privacy preserved record linkage techniques. It is apparent from our results that other than a small trade-off in linkage quality, our framework works better as a combined package of privacy and linkage solution, which any existing frameworks do not yet provide.

Chapter 1

Introduction

Nowadays, many organizations such as healthcare providers, government, or private companies collect huge amounts of data and preserve these data in their databases. Later, these data are processed and analyzed to discover knowledge and interesting patterns that support proper decision making for the betterment of the organizations [41, 73, 98]. In this era of big data, these databases often contain millions, even billions of records. Developing efficient techniques for processing, analyzing, and discovering knowledge from these databases has gained high importance in both academia and industry [130, 149, 171].

The integration of data from the databases of multiple organizations is required for conducting data analysis and improving the quality of decision making [73, 155]. Data integration enriches the data and also improves its quality by detecting duplicate records that refer to the same entity of real-world [58, 77, 187]. An entity, represented by a record in a database, can be a person, a patient, a product, or any other object of the real world. The process of matching and aggregating records that relate to the same entity from different data sources is known as record linkage, entity resolution, duplicate detection, or data matching [58, 61]. Performing record linkage in the context of privacy preservation is known as privacy-preserving record linkage (PPRL) which gains vast attention of the researchers in recent days [45, 183].

This thesis will focus on different aspects of record linkage and, in particular, privacy-preserving record linkage such as data cleaning, scalability, privacy preservation, etc. In this chapter, we provide an introduction to the research presented in this thesis. We describe

the background and application areas of privacy-preserving record linkage (PPRL) using the traditional batch linkage approach and incremental approach in Section 1.1. We then describe the research problem addressed by this thesis in Section 1.2. Our contributions to the research problems are discussed in Section 1.3. Finally, the organization of the thesis is presented in Section 1.4.

1.1 Background and Motivation

The process of identifying record pairs from various databases that belong to same entity in the real-world is called record linkage [61, 187]. Typical applications of record linkage include data warehousing and business intelligence, healthcare systems, master data management, historical research, census, fraud detection, etc. [71, 92]. A primary concern for performing efficient record linkage is that real-world data, especially, health data, suffer from the missing data issues [46, 76]. Record linkage faces two additional challenges that deal with big data as follows. Firstly, the high velocity of updates makes prior linkage results outdated quickly. Secondly, the massive volume of data costs much time for performing a record linkage. These two challenges necessitate an incremental solution so that as soon as data are updated, linkage results can also be rapidly updated [67].

Privacy is another primary concern when record linkage is performed for highly sensitive data, e.g., health records, financial records, etc. [102, 139, 183, 191]. The linkage of records without disclosing identifying attributes of the individuals is known as privacy-preserved record linkage/privacy-preserving record linkage (PPRL), linkage of blind data, or the private linkage of records. PPRL is necessary for linking protected health information (PHI) and has been extensively studied by researchers in recent times [12, 29, 110, 113].

Solving record linkage problems with an incremental approach is comparatively a new research area. Few works have been found in the literature for incremental record linkage, and the available ones only focus on the linkage quality, which is measured by the similarity of resulting clusters, or time efficiency [38, 55, 67, 126, 175, 192, 193]. Still, to the best of our knowledge, there exists no research that addresses the issue of privacy preservation for incremental record linkage. Hence it is interesting to know whether the benefits of incre-

mental record linkage can also be achieved in the privacy preservation context. Moreover, the quality of record linkage largely depends on the quality of datasets, i.e., noiseless and complete data will produce better linkage results [46, 76, 140]. Proper imputation of missing data will help to improve the performance of privacy preserved incremental record linkage framework.

1.1.1 Application Area

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

In public health research, researchers often requires investigating the categories of injuries resulted by car accidents, for uncovering the correlation between accident category and resulting injury [185]. This kind of research can have a great influence on potentially lifesaving shifts in policy-making. In this scenario, several parties such as hospitals, health insurers, police, accident research centers are involved.

The government census agency collects various data from the citizens and the economy of a country. Later, the collected information is used to generate various types of statistical reports by the government. Generally, each census is collected at separate time, saved into a different database of similar structure, commonly contains individuals data regarding age, birthplace, gender, birth year, race, academic qualification, etc. Integration of these data and entity resolution are essential for identifying the characteristics of a population [63].

Financial organizations such as online marketplaces, e-commerce sites, banks require to develop a complete and up to date profile of their customers by linking data from different organizations. Here also several financial institutions such as banks and e-commerce sites are involved [62].

In health surveillance, early outbreak detection systems need health related data from various sources to be gathered and linked, e.g., human, animal, and drug consumption data, for preventing infectious diseases. For linking and storing such data at a central repository, privacy becomes a major concern [183], [109].

1.2 Research Problem and Methodology

Following research problems have been investigated in this thesis:

- Collection of sufficient real health data from heterogeneous sources such as Government hospitals, private clinics, diagnostic centers, specialized institutes, etc. is a difficult task. For the case of most healthcare providers, data are stored manually. Digitization of the data is a challenge. Maintaining the privacy of these sensitive data is another challenge. Different types of attributes of health dataset, their domains, and types, e.g., categorical, ordinal, numeric, etc. need to be studied in depth.
- Develop algorithms to impute missing values for binary, nominal, and numeric health attributes with higher accuracy, precision, and F-measure. Widely used algorithms for missing data imputation will be studied, such as the Fuzzy Unordered Rule Induction Algorithm (FURIA), PMM, and LOGREG version of multiple imputations by chained equations (MICE) algorithms. To achieve better accuracy, precision, recall, and F-measure, improvement of the state of the art algorithms will be performed.
- Develop algorithms to provide privacy to the sensitive medical records of the patients using different encoding and anonymization techniques so that the linkage results can survive better during different privacy attacks. For addressing the current problems with privacy-preserved record linkage, linkage within multiple parties using Honest but Curious (HBC) and Malicious models will be studied in detail. To achieve scalability and enforce privacy, different algorithmic tools, such as generalization, phonetic encoding, etc. will also be studied. For better scalability and privacy of the health records, the widely used phonetic algorithm, Soundex, will be improved. Privacy preservation of the final result will be validated through performing frequency attack, dictionary attack, and calculating entropy.
- Develop algorithms to integrate heterogeneous health records from diverse sources maintaining proper record linkage that will reduce the correlation penalty and DB-Index penalty. For record linkage, different types of clustering methods, such as correlation clustering and DB-Index clustering, will be studied. The recent technique,

incremental record linkage, will also be investigated in the privacy preservation context. Efficient blocking and clustering techniques for health records will be developed in the context of privacy preserved data.

- Develop a framework for incremental record linkage with privacy preservation to reduce linkage time significantly and to support dynamic linkage. Finally, the framework will be implemented to verify the performance of privacy preserved incremental record linkage (PPiRL). Quality of incremental clustering results need to be assessed using Inter-Cluster Similarity and Intra-Cluster Similarity indices, and time efficiency needs to be compared with that of batch record linkage. Privacy preservation of PPiRL needs to be compared with that of privacy preserved record linkage(PPRL).

1.3 Contributions

This thesis focused on PPRL techniques. Specifically, it proposes new algorithms for scalable and efficient PPRL of noisy data, addressing several gaps in existing PPRL research. We have the following key contributions.

1. We propose an improved missing data imputation algorithm SICE, which performs better than existing univariate and multivariate imputation methods for numeric and binary attributes.
2. We propose a novel phonetic algorithm nameGist, which is the only algorithm that supports both English and Bengali name matching. Our proposed algorithm performs significantly better than existing phonetic algorithms and can efficiently process English phonetic names, Bengali phonetic names in English representation, Bengali Unicode names, and mixed names.
3. We propose an improved PPRL technique, Key-based Secured Record Linkage (KSRL). Empirical results show that KSRL can effectively connect records in the scarcity of universal ID numbers and the availability of erroneous data, e.g., misspelled of patient Name. We have categorized the patient identifiable attributes into three categories: changeable attributes, fixed unambiguous attributes, and fixed ambiguous attributes.

4. We are the first to recognize Privacy-Preserving Incremental Record Linkage (PPiRL) as a new field of research. Recognition of this field paves the way for solving the problems of record linkage relating to volume and velocity of data along with privacy issues. We have also formally defined PPiRL.
5. We propose a new end-to-end Privacy-Preserving Incremental Record Linkage framework that encompasses both the privacy and linkage of data. we have implemented our PPiRL framework and compare with traditional privacy-preserving record linkage (PPRL) techniques and incremental record linkage (IRL) techniques. We provide evaluations for the record linkage quality of PPiRL as well as the privacy preservation. It can be derived from our experiments that it is possible to maintain privacy while applying incremental updates on large scale record linkage projects.

1.4 Organization of the Thesis

We present the background of record linkage, privacy-preserving record linkage, and privacy issues of healthcare data in Chapter 2. In Chapter 3, we discuss different techniques for the imputation of missing data and proposed an improved imputation technique namely SICE. Chapter 4 describes our developed phonetic algorithm "nameGist" to support record linkage. In Chapter 5, we present KSRL, a key-based record linkage algorithm that can work effectively in a constrained environment such as the absence of a unique identifier and presence of noise.

In Chapter 6, we formally define the problem of privacy-preserving incremental record linkage. Our developed framework, "PPiRL," which supports privacy-preserving record linkage using an incremental approach, is presented in the chapter. Finally, we conclude the thesis by summarizing our findings and discussing future research directions in Chapter 7.

Chapter 2

Background Study

In this chapter, we provide the background for understanding the concepts of record linkage in Section 2.1. Then we provide an overview of privacy-preserving record linkage (PPRL) in Section 2.2. Different aspects of PPRL are also discussed in the same section. Security and privacy issues related to healthcare data and health information systems are discussed in Section 2.3. The summary of this chapter is presented in Section 2.4.

2.1 Record Linkage

Record linkage is the task of finding same entity from different data sources. It can be presented as a classification problem where record pairs from multiple databases are classified as match if the records refer to the same entity, or as non-match if they do not belong to same the entity [43]. Linking records from multiple databases is not a difficult task if the respective databases have common identifiers. In practice, common entity identifiers e.g., customer-id, patient-id, nationality-id are not available, most of the cases, in the databases to be linked. In these situations, correct record linkage become a challenging task and common quasi-identifiers (QID) i.e., name, gender, date of birth are used for identifying the correctness of the matching records [199].

Figure 2.1 shows the main steps of the record linkage process [58], [183], [184]. The first step of the linking process is known as data pre-processing that includes data cleaning by missing data imputation, noise reduction and transforming data into consistent format. These tasks are vital for better linkage quality as real-world data normally found to be incomplete

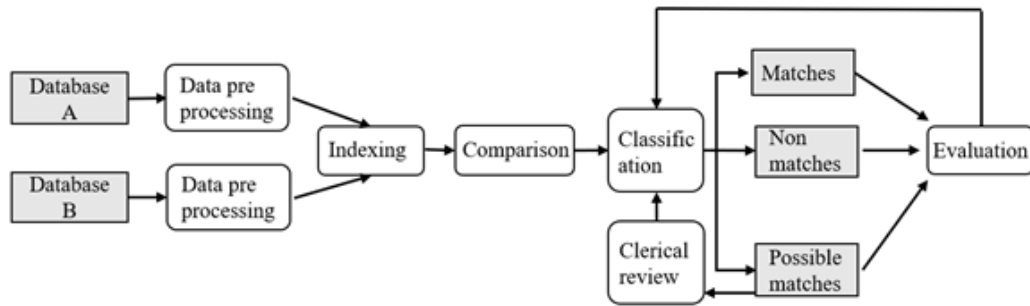


Figure 2.1: Outline of the general record linkage process

and noisy [14] [150]. The second step is blocking that reduces the number of comparisons needed by removing all such records those have least possibility to match. Only those record pairs that have a good probability of matching, known as candidate record pairs, will pass to the next step for detail comparison.

In the third step, candidate record pairs are compared in detail using different similarity functions. Usually, multiple attributes (i.e., QIDs) of candidate records are used to compare matching which output a vector containing similarity values of the QIDs.

In the fourth step, the similarity vectors of the previous step are inputted to some decision model which then classifies the record pairs into matches and non-matches. The third type of classification named possible-matches is also done when the classification model cannot make a final decision about matching. The record pairs, classified as possible matches, gone through a time-consuming manual review process which finally decides them as matches or non-matches.

In the last step of record linkage, the linkage quality, completeness, and complexity are evaluated before external applications can use the record linkage results. In the next subsections, the preprocessing, blocking, comparison, and classification steps of record linkage process are discussed more details.

2.1.1 Data Preprocessing

Preprocessing of data helps improve the condition of data by handling errors and inconsistencies from data. Although data quality issues are found in a single dataset, quality issues become serious when data is integrated from multiple sources into a warehouse [150]. Some

essential steps of data pre-processing are listed below.

Handling missing data: Real world dataset always suffers with missing data. For many reasons a dataset can be incomplete and some vital data can be missing. The situation is even worse for the case of healthcare research. A dataset with missing data may led to wrong conclusion or misleading prediction by data mining algorithms. So, handling missing data is an important part of data pre-processing [27]. Details of missing data imputation is discussed in Chapter 3.

Data Cleaning: A common sense is that inputting garbage in data analysis will output garbage. Data cleaning process detect and correct corrupt, noisy or dirty data in a dataset thus improve the overall quality of a dataset [150]. An important part of data cleaning for record linkage applications is the misspelling of names in real datasets which causes a single or same person to identify wrongly. Phonetic encoding algorithms help in this regard to remove noise in name matching or entity resolution [101]. Details of phonetic encoding is presented in Chapter 4.

Feature selection: Feature selection is a process of selecting a subset of the total features according to specific criteria. It is an essential and frequently used technique in data analytics for dimensionality reduction. It reduces the number of features by removing irrelevant and redundant attributes. Various studies show that a good number of features from a dataset can be removed without performance deterioration of the data analysis. It has many positive impacts, such as speeding up data mining algorithms, improving accuracy, and help to build a comprehensive prediction model [132], [205].

Standardization: Data standardization is a vital part of ensuring data quality. Lack of standardization will result in defective data that has numerous adverse effects. Standardization includes transforming data into consistent and well-defined forms, resolving inconsistencies in data representations, and necessary encoding [41].

2.1.2 Blocking

Blocking is an essential step for record linkage process from practical viewpoint. If we link two databases D_1 and D_2 , contains n_1 and n_2 records respectively, then it requires $n_1 \times n_2$ number of comparisons. For large databases, this is infeasible. Comparing all records of both

databases is also unnecessary as majority of them will be found as non-matches. Blocking step help the record linkage process to filter the unnecessary and infeasible task of comparing each pair of records. Blocking is also refereed as indexing or searching technique [42].

Blocking key is a single attribute, or may be a combination of attributes used to chose the block or cluster to insert a record. Records with the same blocking key value will be inserted into the same block. Candidate record pairs are generated from records within the same block. In the next step, known as comparison step, these candidate record pairs are compared in detail.

Blocking made a trade-off between the correctness of the record linkage and the computational complexity. If a less specific blocking key is chosen, it will produce larger blocks with more candidate record pairs. This will increase the chance of finding more true matches, but additional computation cost will also be needed. On the other hand, a more specific blocking key will generate many small blocks that eventually reduce the computation cost. However, this will increase the chance of overlooking some correct matches [15].

Researchers have presented many blocking techniques for data matching and record linkage. Christen P. [42] and Nin J. et al. [136] presented a survey on popular blocking or indexing techniques used in data integration and record linkage. In the pioneering research of record linkage [61], the standard blocking idea was used where all records with the same blocking key value were inserted into the same block. Later, in the comparison step, only the same block records were compared. This approach reduces the total number of comparisons to $(n_1 \times n_2) / b$, where b is the size of the block. Some widely used blocking techniques are discussed briefly below.

In traditional blocking, one attribute or a combination of features is used for indexing or grouping similar records from a dataset. For example, if, in a dataset, "Postcode" is used as a blocking key, then, each generated block will contain only those records that have the same postcode. It helps to avoid comparison of all records in a dataset and reduces the comparison space. The attributes used for blocking are known as blocking keys (BK). In our research, we have used traditional blocking technique which is further explained in Section 6.4.3.

Mapping Based blocking is also a popular indexing technique [92]. This technique is used to convert blocking key values into objects that are mapped into a multidimensional

Euclidean space. Then multidimensional similarity join is applied to the same cluster to group similar objects. This approach is modified in [1] by using two levels of mapping. The first level of mapping is identical to [92] where blocking key values are mapped into a multidimensional Euclidean space, and the second level is lower-dimensional metric space using edit distance.

Hashing based indexing is also a widely used technique introduced in [64] to solve the problem of high dimensionality. This indexing aims to speed up the similarity search in the approximate nearest neighbour problem. Locality sensitive hashing (LSH) is considering one of the popular hashing approaches that are used to address objects with high dimensionality citedatar2004locality. It uses LSH functions to hash records where the values of attributes are used to convert into a set of binary numbers. Then these patterns are used to group records into similar blocks according to their hashing values.

q-gram-based blocking techniques, useful for low-quality data, insert a record into multiple blocks by producing variations of blocking key value using q-grams [31], [42]. Suffix array-based blocking methods are similar to the q-gram-based approach where blocking key values are used to generate suffixes, and later blocks are extracted from the sorted suffix string array [54]. Another recently proposed blocking technique named HARRA supports similar values most likely to be hashed into the same block [111]. The algorithm presented in the paper performs blocking much faster than its competitors with scalability support.

2.1.3 Comparison

There are mainly two types of comparison techniques: exact and approximate. In exact comparison, a function measures whether the attribute values of two records are equal or different. It is a straightforward approach but is not suitable in many real-world scenarios. Contrarily, in approximate comparison, a function measures how similar are the attribute values of two records. As real-world datasets contain many typographical errors and noises, later approach is better for practical applications [75].

Comparing two entities or records can be done either at attribute (feature) level or record level [144]. In the case of attribute level comparison, the similarity is checked between attribute or feature values of the comparing records using special comparison functions de-

pending on attribute type. On the other hand, record level comparison approach merged values of all the features of a record into a single long string, then compare these strings of the candidate records.

Approximate matching functions express similarity on a numerical scale. Normally a 1 is assigned for complete similarity, a 0 for complete dissimilarity, and some value between 0 to 1 for partial similarity. In [135], popular approximate matching algorithms have been surveyed. Selected algorithms that are widely used are presented below.

For record level comparison, SoftTF-IDF [49] string comparison technique can be used. It can compare strings of several words using the concepts of Term Frequency (TF) and Inverse Document Frequency (IDF) [152]. Like information retrieval, it calculates weights to words according to their total occurrence in a dataset and the similarity of two strings is measured as the highest similarity-value between word pairs in the strings.

In record linkage applications where entity names and their addresses are required to be compared, a good choice is Jaro-Winkler method [91], [195]. By using the expertise obtained by conducting large-scale linkage projects, Jaro-Winkler approach was developed at the US Bureau of the Census around 1990. M. Jaro combines edit distance and q-gram-based approach. Later, W. Winkler improves Jaro's basic comparison function e.g., weight adjustments based on the lengths of two strings.

A popular comparison method, Levenshtein (edit) distance [120] calculates the smallest number of edit operations needed to convert a string to another one. Edit operations include character insertion, deletion and substitution. Few extensions of the basic edit distance algorithm have been developed i.e., setting different costs for different types of edit operations. Another popular comparison function that uses the idea of comparing common sub-strings between two strings, is known as q-grams [180], [31]. The candidate strings split into shorter sub-strings of "q" length by sliding window technique. Then the number of q-grams, common to the both strings, is counted.

2.1.4 Classification

In record linkage, after the comparison step, usually, a decision model is used to classify the records as matches, non-matches, and possible matches. There are different types of

classification models, e.g., probabilistic, rule-based, threshold-based, and machine learning-based models. They are described in brief below.

The probabilistic model is widely used for record linkage classification. It was first proposed in [61]. Here, the possibility that two records matches or not is modeled using prior error estimates in the data and the frequency distributions of individual attribute values. The approximate similarity of the candidate records calculated in the comparison step is also considered. Later, W. Winkler proposed some improvement of the basic model in [195] and [196].

A threshold value is used in the threshold-based classification model to classify the record pairs [70], [40]. The threshold value, calculated as $T = \sum_{i=1}^k S_i$, is the summation of the overall similarity values of the candidate pairs, for each candidate record pair. This threshold value is then used to determine into which class the record pair belongs.

Recently both supervised and unsupervised machine learning approaches are being used in classification models for better accuracy [58]. Supervised approaches used correctly labeled training data to train the decision model. Later, the model can classify un-labeled record pairs. Support Vector Machine (SVM), k-Nearest Neighbors (k-NN), and decision trees are some popular supervised learning algorithms used in record linkage [167], [19], [40]. A problem faced by supervised techniques is that the required amount of training data is not always available, especially for the case of sensitive data e.g., medical, financial data.

On the other hand, unsupervised machine learning models do not need a training dataset to classify record pairs. One popular unsupervised technique is clustering. It groups similar record pairs such a way that each cluster contains the records that refer to a real-world entity [127]. For taking the final decision about the potential matched records, a clerical review is required. In this case, a semi-supervised learning technique known as active learning, can be used for manual classification [8].

2.1.5 Evaluation

The last step of a record linkage process is to evaluate its quality and efficiency. The linkage quality is commonly measured by using quality metrics such as accuracy, precision, recall, and F-measure. Precision, recall and F-measure are more suitable for measuring linkage

quality than accuracy in all situations [183]. Accuracy is not always a good matrix for quality measure as record linkage is normally an imbalanced classification problem. For example, the number of non-matching record pairs is significantly higher than the number of matching pairs which can highly affect the accuracy value [43].

The efficiency can be measured by analyzing the scalability of a linkage technique on large-scale real-life applications with millions of records. Scalability of a record linkage application can be evaluated using measures based on the ratio of candidate record pairs and also the measures that depends on computing resources and networking infrastructure [42].

2.2 Privacy Preserving Record Linkage

Nowadays, maintaining privacy and confidentiality are significant challenges for record linkage. There are mainly two reasons for this increasing demand for privacy. First, record linkage applications are now widely used in sensitive projects where maintaining privacy and security is a fundamental need, e.g., health sectors, banking sectors, e-commerce applications. Second, security and privacy vulnerabilities, new threats, hacking, data breaching are in the highest pick in the history of humankind. [109]. During the linking of databases across organizations using personal information, careful protection of the privacy of this information is a must.

Different organizations' databases required to be linked in such ways that sensitive data is not revealed to any of the involved parties in any cross-organizational project. In addition, no adversary might be able to learn anything about the sensitive data in cross-organizational record linkage. The process of discovering records of similar individuals from different databases without disclosing identifying attributes of these individuals is known as privacy preserving linkage of records, linkage of blind data, private linkage of records or in short PPRL [188], [185]. The formal definition of PPRL is presented next.

Definition 2.1. Privacy-preserving record linkage (PPRL): Let P_1, \dots, P_m are the m owners of the databases D_1, \dots, D_m , respectively. They wish to find out which of their records $R_1^i \in D_1, R_2^j \in D_2, \dots, R_m^k \in D_m$ match based on their demographic data according to the decision model $C(R_1^i, R_2^j, \dots, R_m^k)$ that classifies records of different dataset

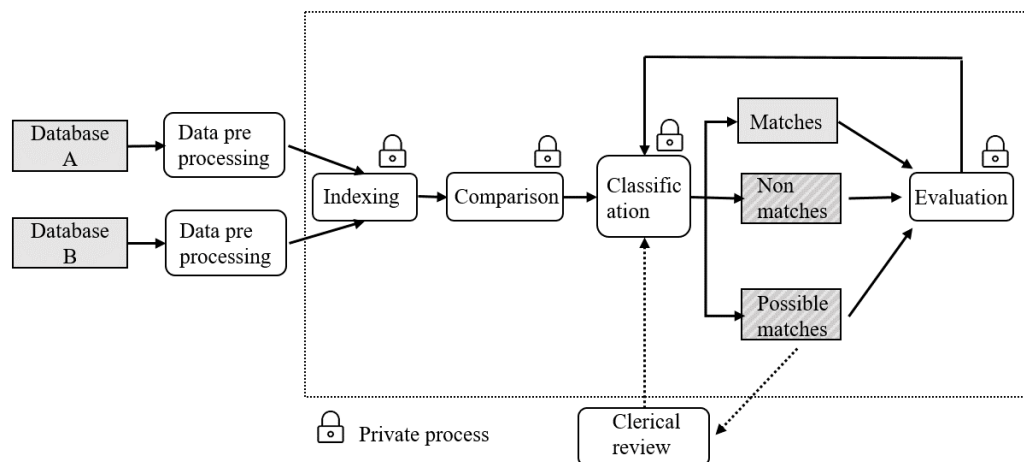


Figure 2.2: Outline of privacy preserved record linkage (PPRL) process adopted from [183]

into following classes: M (Match), and N(Non-match). P_1, \dots, P_m wish to preserve the privacy of their actual records $R_1^i, R_2^j, \dots, R_m^k$ from other parties. However, they are ready to disclose the actual values of some selected attributes of the records to each other, or to an external party, those are in class M to allow analysis [183]. The essential properties for a PPRL solution workable in real-world applications are linkage quality, scalability, and privacy.

Privacy of records needs to be considered in all the steps of a PPRL process, which makes the cross-organizational linkage more difficult. The PPRL process is depicted in Figure 2.2. Data pre-processing can be done independently at the participating data sources. For that reason, generally, pre-processing is not considered a part of the PPRL process. All data sources need to conduct the data pre-processing the same way on the data they will use for linking. So, information exchange is required about what data pre-processing methods the parties are using and the list of attributes they have in common for record linkage.

Blocking in PPRL required to be conducted in such a way that the sensitive information of one party which would allow inferring individual records in the databases can not be revealed to other party or an external adversary. Values of the selected attributes, used for comparing records among exchanging parties, often contain noises and typographical errors, and therefore just encoding these values with a encryption technique and comparing the encrypted values will lead to a poor linkage quality. For a small variation in attribute value may lead to a different encoded value. So, secure and efficient calculation of the approximate

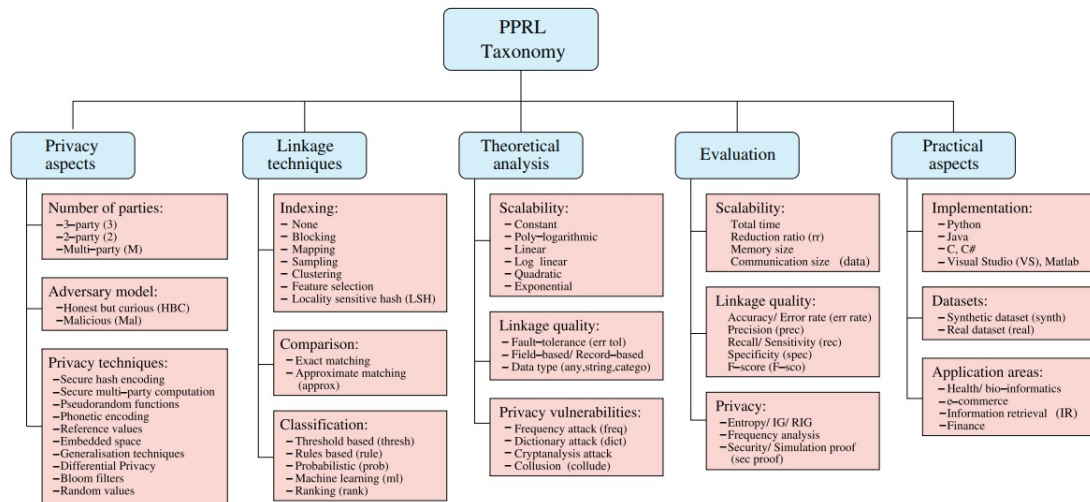


Figure 2.3: A taxonomy of PPRL techniques adopted from [183]

matching of attribute values is required.

In the context of PPRL, classification needs to be performed in such a way that no party could be able to learn any information about the non-matched records in the databases of other parties. The information may be similarity values for specific attributes of individual record pairs, or a distribution of similarity values across the candidate record pairs. The evaluation of the quality of linkage in the context of privacy-preservation is also a challenging task. This is due to the fact that, in PPRL, access to the actual record values may be impossible as it will reveal confidential information about the records.

An useful taxonomy of PPRL is presented by Vatsalan et. al. in their paper [183] where they characterize PPRL techniques along 15 dimensions. Many of these dimensions will be covered in this thesis in various places. Figure 2.3 presents their proposed taxonomy. The following subsections provide more detail of different privacy aspects for privacy preserving record linkage.

2.2.1 Privacy Techniques

A lot of privacy techniques has been proposed by the researchers to ensure privacy preservation in PPRL. Selected privacy techniques have been presented below.

- (a) **Phonetic encoding:** A phonetic encoding algorithm, e.g., Soundex, Metaphone, nameGist groups values together according to similarity of pronunciation [101, 147, 156]. Phonetic encoding algorithms inherently provide privacy [96, 183]. They also improve

scalability by reducing the number of comparisons [39]. Another advantage of phonetic algorithms is that they support approximate matching by handling typographical errors [39, 96]. A notable limitation of these algorithms is that they are language-dependent, and only limited work has been done on multi-lingual phonetic encoding [101, 169]. Details about phonetic encoding are presented in Chapter 4.

- (b) **Generalization techniques:** To overcome the risk of re-identification of the entities, generalization techniques are used. They generalize the data in such a way that re-identification from the generalized data is not possible [72, 174]. A widely used generalization technique for privacy preservation is k-anonymity. A database will satisfy the k-anonymity criteria if every combination of quasi-identifiers of the database will be shared by at least k records [95, 119, 122]. We have used generalization techniques, especially k-anonymity, as a part of privacy preservation in our proposed PPRL techniques, presented in Chapter 5 and Chapter 6.
- (c) **Secure hash encoding:** One-way hash functions convert a string into a hash-code such that analyzing only the hashed value will make it impossible with available computing technology to find out the original string [128]. For example the hash value of the string "Bangladesh" is "f78a77f631d275aac6a914a17fe1b885" using MD5, a popular hashing algorithm. Hash encoding is one of the oldest methods for privacy preservation [57, 93]. Secure Hash Algorithms, e.g., SHA-1, SHA-256, and Message Digest, e.g., MD4, MD5, are the widely used hash algorithms. A considerable problem of using hash functions for privacy preservation in record linkage is that a single character difference in the input string will produce a completely different hash-code. So hashing is suitable for exact matching based record linkage. We have used the hash algorithm MD5 in Chapter 5 for a part of privacy preservation.
- (d) **Secure multi-party computation (SMC):** The underlying idea of SMC computation is that if at the end of the computation, no party knows anything except its input and the final results, then the computation can be marked as "secure" [123, 201]. Micali et al. developed a general framework for SMC, applicable to multiple parties [129]. Commonly used SMC techniques are secure set union, secure set intersection, and

secure scalar product. SMC techniques are computationally expensive, which is an inconvenience of using them in PPRL. However, they can be used in record linkage [59].

- (e) Reference values: Several PPRL techniques use the reference value approach for privacy preservation [95, 141, 198]. Here, some values, common to all database owners or exchanging parties, can be used to enforce privacy. For example, a public telephone directory can be used for this purpose. The exchanging parties will then calculate the distances between their attribute values and the reference values.
- (f) Noise addition: It is a data perturbation method that works by Adding noise in the form of additional records to the databases that are used for linkage [97]. Noise addition can successfully overcome the frequency analysis attacks with a cost of degradation in the accuracy and scalability of record linkage [96].
- (g) Bloom filter: It is first proposed by Bloom for checking a set membership efficiently in 1970 [21]. Recently, bloom filters have been used in PPRL for matching records privately [56, 116, 170]. A Bloom filter is a bit-string data structure of length m bits. Initially, all the bits of a filter are set to 0. A k independent hash functions, h_1, h_2, \dots, h_k , each with range $1, \dots, m$, is used to map each of the elements in a set into the Bloom filter by setting respective k bit positions to 1.

2.2.2 Privacy Attacks

There is no such thing as "True Security" or "True Privacy". With the increasing techniques of privacy over time, different and new attacks have been formed to violate the privacy techniques. The main attacks and vulnerabilities of PPRL techniques are briefly described below.

- (a) Dictionary attack: A dictionary attack is usually carried out to break passwords or similar encrypted data in a database with the help of available digital dictionaries [133]. To carry out a successful dictionary attack, a hacker must have access to a dictionary or list of frequently used words or vocabularies. General dictionaries are useful in this matter as they provide millions of words that could be used to create a password for a user. As a dictionary contains so many words that it generally contains peoples usable

passwords and hackers used a technique to go through all the word on a dictionary and find a word to decrypt the data.

- (b) Frequency attack: Frequency attacks used the frequency distribution of a set of masked values in comparison with the frequency distribution of a set of known plain-text values [3, 94]. Frequency distribution of the characters of specific attributes in a dataset may cause information exposure. If the frequency of letters remains the same, even after applying privacy techniques, then it is possible for the attackers to re-identify by matching the frequency distribution. Hackers do not need any prior knowledge about the encoding process, the only knowledge needed is which attributes are encoded and have access to a public database of these attribute values and their frequencies from the same domain [44].
- (c) Collusion In multi-party protocols: Dishonesty or collusion by some of the parties among the participating parties may make the privacy of a PPRL technique vulnerable [47, 185]. Recently in multi-party PPRL protocols, collusive behavior was observed by some of the participating parties. They were trying to gain access to unauthorized data in different application domains, such as online rating, auctioning, and mobile computing [5, 24]. Collusion aims to learn the sensitive data of another not colluding party by colluding parties sharing their data and parameter settings. Information entropy technique may be used to gain knowledge from sensitive privacy preserved data of a party.

We have performed the privacy evaluation of our proposed PPRL methods using frequency analysis, dictionary attack, and information gain. Details have been discussed in Section 5.3.2 and Section 6.5.4.

2.2.3 Stakeholders in PPRL

Multiple stakeholders or parties participate in the record linkage process of PPRL, depending on the linkage model. They are mentioned below.

- Database owner or Exchanging Parties: The exchanging parties are the database owners or data custodians who want their databases to be linked [41, 142]. The database

owners may directly participate in the record matching or comparison step or may send their dataset to a third party (known as a linkage unit) to perform the linkage. It depends on the predefined record linkage model, upon which the exchanging parties agreed initially. Healthcare providers, banks, or other financial institutions, different divisions of government are examples of database owners.

- **Record Linkage unit:** A linkage unit is a special entity, usually a third party that participates in the linkage process, based on the linkage model chosen. In the context of PPRL, it is a common scenario that the database owners send their datasets to the linkage unit. It then links the records, ensuring privacy, and share the matched records with the owners or exchanging parties [153, 183].
- **Consumers:** Consumers are the users of the linkage data. After the completion of the record linkage process, the matched results are sent to the consumers. Examples of consumers other than the data exchanging parties include data analysts, external researchers, the Government [41, 47].

2.2.4 Adversary Models

The commonly used adversary models used by the database owners and record linkage units in the context of PPRL are: Honest but curious and Malicious model. Moreover, the Covert advisory model is also used in few researches. They are described briefly below.

- **Honest-but-curious (HBC) adversary model:** Majority PPRL protocols, proposed by various researchers, follow the HBC or semi-honest adversary model [4, 143, 183]. It is also known as a passive adversary model. In this model, all the stakeholders of the record linkage process follow the steps of the protocol. However, the stakeholders are curious to learn some information about other. During record linkage, a database owner or the record linkage unit may preserve the results it received from other owners. Later, these results may be used to infer sensitive attribute values of a database of other parties using the frequency analysis technique [44, 114]. In Chapter 5 and Chapter 6, we consider HBC model for our experiments.

- **Malicious adversary model:** This model primarily assumes that the parties involved in the protocol may not follow the steps of the protocol; instead, they may behave whimsically [123, 131]. A malicious database owner may send malicious data to another owner to get sensitive information from that owner. Generally, secure multi-party computation (SMC) technique is used for the PPRL process in the context of a malicious adversary model [201]. SMC uses encryption and encoding techniques to ensure that no party can learn any sensitive information from other parties. However, SMC adds additional communication and computation complexities compared to the HBC model.
- **Covert adversary:** The covert adversary model is a trade-off between HBC and the malicious models. Here it is assumed that a party may deviate arbitrarily from the specification of the protocol in an attempt to cheat until they are being caught. A database owner, who is cheating, can be caught by an honest owner with a preset probability, which is called the deterrence factor. The covert adversary protocol is useful in many real-world scenarios where the assumption of the HBC model does not suffice, and the malicious model is expensive to achieve [10, 123].

As this thesis particularly focuses on record linkage of health data, we will further investigate the security and privacy issues of health data and health data servers in the next section.

2.3 Health Records and Health Data Server: Privacy and Security Issues

In this section, we have presented a comprehensive review of the security and privacy risks of digital health data and integrated health information systems. We have discussed the statistics of the high rise of security threads in healthcare data servers. Health data or health records refer to pieces of information collected to diagnosis a health condition. A health record is collected about a patient, his family, often during the creation of a nursing history for the patient. A health record may include multiple types of health data such as various notes

entered by health care professionals over time, recording observations and administration of drugs, test results, x-rays, reports, etc. Digital health data are health data generated by medical devices in the digital form, e.g., fasting plasma glucose test result, or other health-related information, e.g., height, weight, blood group, etc. stored in digital format in computers, laptops, or in a database of health information systems [25, 154, 203].

At present, large amount of digital health data are generated daily by healthcare providers. Medical records of patients are increasingly stored in digital form, such as Electronic Health Record (EHR). EHRs are more useful than paper records for better healthcare and medical research because electronic data can be stored easily and manipulated by software. These precious data are stored in various health information systems (HIS) in hospitals, research centers and diagnostic laboratories. Some attributes of these health records falls in the category of protected health information (PHI) [18, 84].

PHI is defined as personally identifiable health information collected from an individual, and covered under federal or international data breach disclosure laws [137]. PHI of an Individual relates to:

- a. the individuals past, present, or future physical or mental health or condition,
- b. the provision of health care to the individual,
- c. the past, present, or future payment for the provision of health care to the individual

PHI includes common identifiers such as name, date of birth, address, national ID / social security number, telephone and fax numbers, E-mail addresses when they can be associated with the health information listed above [79]. Laboratory reports, medical records, and hospital bills are examples of PHI because each document contains a patients name and/or other identifying information associated with the health data content.

Security of a health information system deals with protecting medical data from intruders, malwares, and frauds. It retains confidentiality and integrity of healthcare data. Privacy concerns exist wherever personally identifiable information or other sensitive information is collected and stored in any form. A major challenge in health data privacy is to share data among medical practitioners while protecting privacy of PHI. Privacy of health record may be applied in many ways, e.g., encryption, authentication, and data masking [74, 166].

Nowadays, hacking PHI by cyber-criminals is observed as a growing trend. Hackers goal is to take advantage of personal information of the patients. The average sale value of a complete medical record varies from \$10 to \$1,000 in the underground market. Although the privacy of a patient can be compromised with paper-based medical records, the chance is highly increased with digitized record-keeping by the healthcare providers [26, 86].

2.3.1 Data Breaches of Health Information Systems

A health data breach or leakage is defined as an event that involves the loss or exposure of personal health records. Personal health records are data containing privileged health related information about an individual that cannot be readily obtained through other public means, which is only known by an individual or by an organization under the terms of a confidentiality agreement [36] . For example, leakage of a health insurers record of the policyholder with doctor and payment information will be treated as a health data breach.

The costs of a data breach may vary incident wise, with respect to place and time. The cost includes the direct and indirect cost. Direct costs refer to the direct expense spend to carry out a given activity such as hiring forensic experts and law firm or offering identity protection services to the victims. Indirect costs include the time, effort and other organizational resources spent during the data breach resolution. Indirect costs also include the loss of goodwill and customer attrition. The average cost of data breach per lost or stolen consumer or service data is 136USD, but in the case of a breach of healthcare organization, the average cost is 363USD [87].

2.3.2 Some Incidents of Health Data Breaches and Factors Behind

According to the report [89], for the first time, criminal attacks are the number one cause of healthcare data breaches. Criminal attacks on healthcare organizations are 1.25 times higher compared to five years ago. The main causes of data breach in healthcare sectors are illustrated in Figure 2.3.

Some recent attacks on health information centers are listed below:

- Hackers have shut down the internal computer system at a Hollywood Presbyterian Medical Center for more than a week for a payoff of 9,000 bitcoins, or almost USD

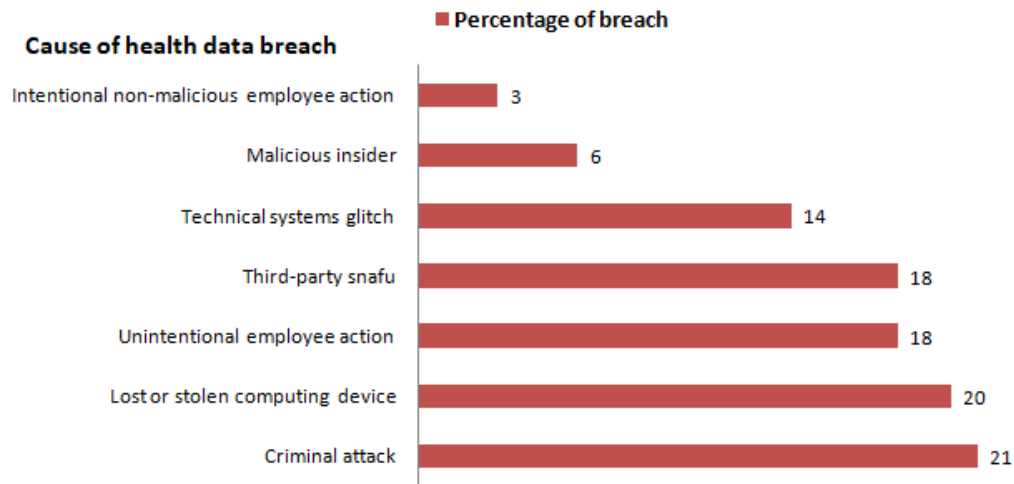


Figure 2.4: Main causes of data breach in the healthcare industry

3.7 million [51]. It is due to a malicious software called ransomware that encrypts sensitive data until it can only be decrypted with a code.

- In February 2016, Jackson Health System discovered that a hospital employee have stolen confidential PHI of patients including names, birthdates, social security numbers and home addresses around 24,000 patient records over the last five years [37].
- Premera Blue Cross was targeted with a sophisticated cyber attack after hackers gained access to the financial and medical information of 11 million members in January 2015. Hackers swiped Social Security numbers, financial information, medical claims data, addresses, email addresses, names and dates of birth [172].
- In last ten years, at least 18 health breach reported in Europe affected minimum 9,337,197 individual records [36]. The health records include details on the patients conditions, names, home addresses and dates of birth. The health networks and servers containing integrated health records are in high risk of cyber attacks all over the world.

2.3.3 Cyber Attacks on Healthcare Servers

From 2014, hackings on healthcare servers increased terrifyingly. The attackers motivation is to get huge PHI in a single successful hack. We have analyzed the data provided by U.S. Department of Health and Human Services and found that hackers are increasingly targeted healthcare servers which is very alarming to national level health information system

development. Table 2.1 presents the fact clearly. We have summarized these data from [78].

Table 2.1: Statistics of healthcare server attack compared to total healthcare breach

Reporting Year	Total Health Data Breach affecting 500 or more individuals	Healthcare Server Attack
Year 2011	194	27
Year 2012	202	25
Year 2013	263	35
Year 2014	290	55
Year 2015	265	50

2.3.4 Other Impacts of Health Data Breaches

There are other impacts of health data breaches. They are discussed below:

- a. Breaches of PHI drastically effect the goodwill of a healthcare organization. In a research report, it is shown that, people are withholding their health information from healthcare providers because they are concerned that there could be a confidentiality breach of their records [186]. An unwillingness to fully disclose information could delay a diagnosis of a communicable disease. This is not only a potential issue for the treatment of a specific patient; there are potential public health implications.
- b. Penalty of healthcare providers are imposed in two ways. They have to pay ransom to the hackers to get their breached data back or to restore their hacked system [51] and they also pay fine to the government for failing to safeguard patient information [34].

2.3.5 Analysis of the Risks of Health Information Systems

It is quite clear that the main reason of the breaches are the sell value of complete health records. What makes medical data so unique is that it often contains most of the information hackers are looking for such as credit card information, and Social Security and bank

account numbers giving them a one-stop stealing strategy. Fraudsters use this data to create fake IDs to buy medical equipment or drugs that can be resold, or they combine a patient number with a false provider number and file made-up claims with insurers. Sometimes, the cyber criminals use this data to blackmail a patient with good social status. For example, the formula one (F1) racing legend Michael Schumachers and pop legend Michael Jacksons medical records were hacked for money [109].

Another important thing to notice is that, a healthcare company is looser in many ways after a successful breach. It has to pay money to both the hackers and the government. This situation will eventually increase healthcare cost and decrease better healthcare delivery.

If the stored health data are de-identified in every place from health information system software to backup and also in health data warehouses then the risk of data breach can be significantly reduced. Because there is almost no sell value of de-identified health records. Another positive thing of de-identification is that, if a data breach occurs, privacy of individual patient will not be affected.

2.4 Constraints of Health Data in Developing Countries

Developing countries are those with low, lower middle or upper middle incomes. There are some common socio-economic characteristics found in the developing countries of the world that have a similar impact on healthcare facilities and health data. These characteristics include Lower per-capita income, higher population growth rates, and low level of urbanization [99, 176]. This implies poor health and inadequate education. In these countries, most of the people live in the rural areas. Above socio-economic conditions made an impact in the available health care data of Bangladesh and other developing countries in the following ways.

- Health records without unique Patient ID: people do not have medical cards with unique health ID. Health care centers do not have provision to store National ID numbers or Social Security Numbers (SSN).
- Misspelled names: Many people in real do not know their full name and unable to pronounce their name correctly even in the mother tongue. The Same person provides

a different version of their name in the health care facilities. The problem can be understood from the Table 2.2.

Table 2.2: Ambiguity in patients' name imputation

Actual Patient Name	Various Inputted Name
Ramim Hossain	Mr. Ramim Hossain
	Ramim Hossain
	Mr. Ramim
	Md. Ramim Hosen
	Mr. Md. Ramim Hossain
	Ramem Hossain
	Romim Hossain
	Ramim Hosen
Ramim Hosain	

- No actual date of birth: Enormous people do not know their actual birth date because of lacking of birth registration. For several years, they provide same age (e.g., 43 years) to hospitals and diagnostic centers. A lot of people do not know their actual date of birth in Bangladesh. This is a very common scenario for aged rural people with less education.
- Missing attribute values: As Bangladesh and other developing countries have the dense population and inadequate facilities, in all health centers, there are long queues of patients. So many necessary attribute values cannot be inputted for processing a high number of patients in limited time.
- Error in data: less qualified staffs for inputting patients data. This leads to unintentional wrong input data.

So patients health records in Bangladesh contain more noisy data with more missing values and without unique patient identification numbers. These limitations make record linkage

methods developed for advanced countries, unsuitable for developing countries. Thus a more specialized technique is needed to address the situation.

2.5 Healthcare Data Generation Scenario in Bangladesh

Health data are generated in different places such as hospitals, diagnostic centers, etc. We have considered the data generation scenario of Bangladesh as a case study. The patient visit cycle to different hospitals, diagnostic centers, and private practitioners chamber is illustrated in Figure 2.5 to Figure 2.8. The patients' visit to different health service providers can be grouped as follows.

2.5.1 Patient treatment cycle in Bangladesh

Patient visits hospitals: There are two types of hospitals in Bangladesh, Government hospitals and private hospitals. According to Directorate General of Health Services (DGHS), the total number of government hospitals under DGHS is 592 [104]. According to the list provided by Bangladesh Private Clinic and Diagnostic Owners Association (BPCDOA), the only Government approved association of private hospital owners, there are 2761 private hospitals in Bangladesh [106].

Patients normally visit a hospitals outdoor or OPD unit, where the person in the reception notes down the basic information of the patient. Then the corresponding doctor checked the patient and write up the treatment notes. If necessary, the doctor gives some pathological tests that the patient performed in the diagnostic unit inside the hospital or any outside diagnostic center. The test results are stored in the centers where a test is performed. In almost all hospitals, there is no patient tracking system with unique patient ID. The irony is that the number of times same patient visits same hospital for treatment or diagnosis, his or her records will be recorded each time as a different patient with different ID or serial number.

Patient visits diagnostic centers: According to Bangladesh Private Clinic and Diagnostic Owners Association (BPCDOA), there are more than 8000 private diagnostic centers in Bangladesh registered by the Government. A patient may visit any diagnostic center to per-

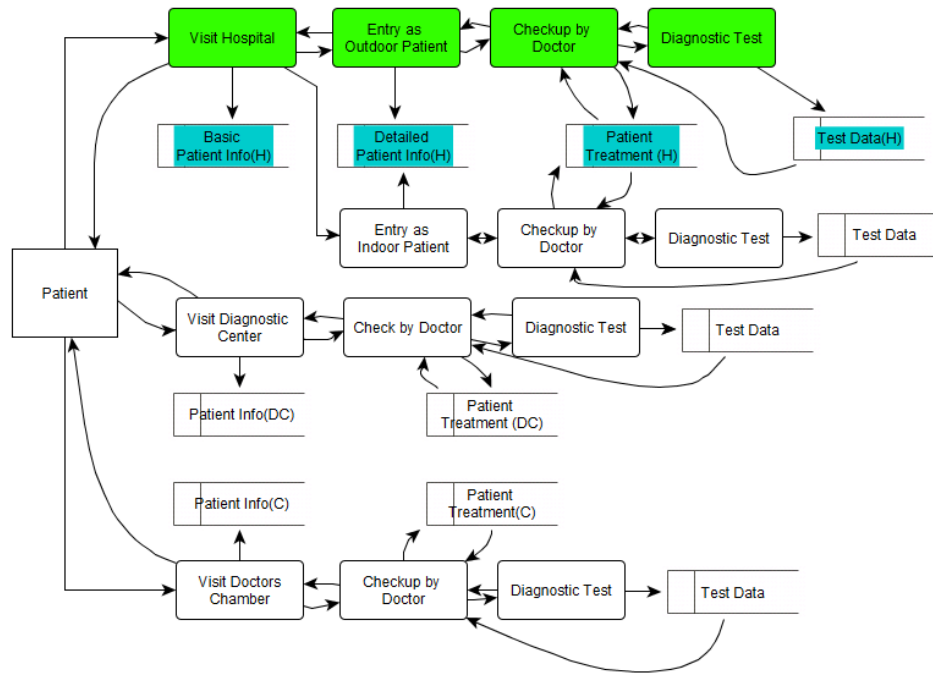


Figure 2.5: Patient treatment in a hospital as an outdoor patient

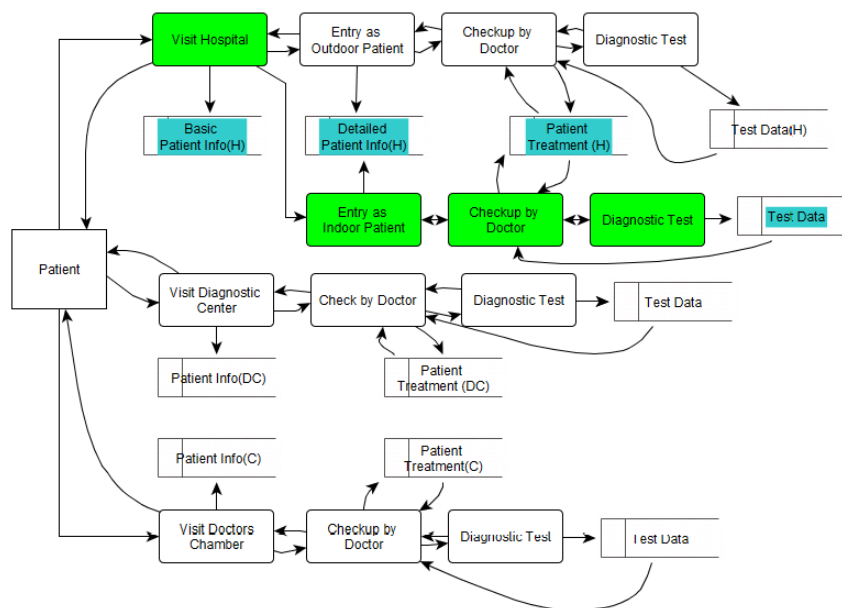


Figure 2.6: Patient treatment in a hospital as an indoor patient

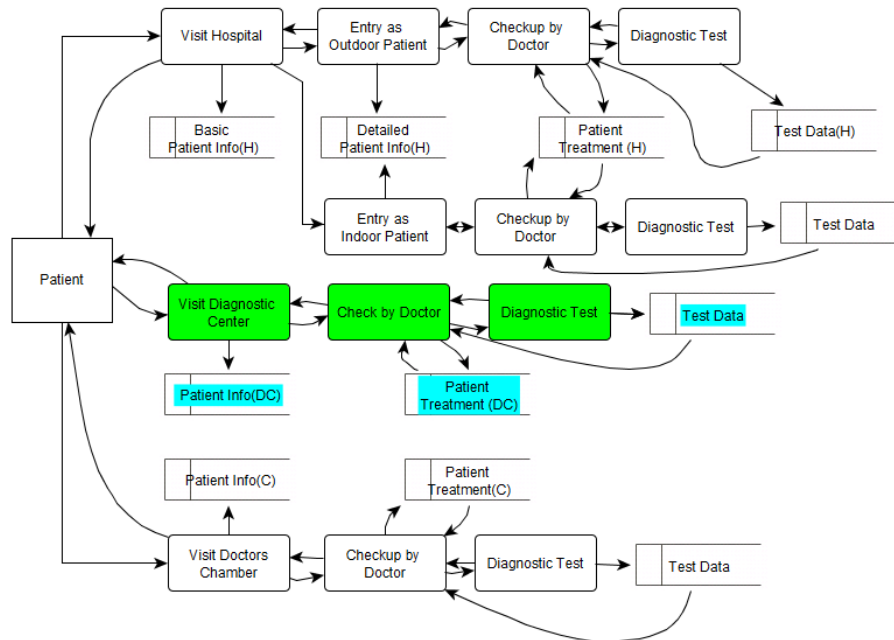


Figure 2.7: Patient treatment in a diagnostic center

form some routine health checkups to know his health conditions. These tests include Blood sugar, Cholesterol level test etc. In almost all Diagnostic Centers (more than 99%), every time when same patient visits, he is treated as a new patient and his records are stored as a new entry with no relationship or linking with the previous records of the same patient. The situation is illustrated in Figure 2.7.

Patient visits personal chamber of doctors: There are about 75700 Registered MBBS doctors and 6800 Dental doctors in Bangladesh [104]. Most of the doctors have private chambers where they consult patients after office hours. A patient can visit a doctor's chamber for treatment. The doctor may recommend some pathological tests. Here also the patients are not tracked with unique ID and no linkage is maintained among the test records of a single patient. The situation is illustrated in Figure 2.8.

2.5.2 Record Linkage Problem

Based on the patient cycles as described above, different cases arise.

Let us denote a patient by P, healthcare center by H, timestamp by T, and event by e.

Case 1: Patient P1 visits hospital H1 at timestamp T1 for event e1 with ID concat(P1H1T1e1)

Case 2: Patient P1 visits hospital H1 at timestamp T2 for event e1 with ID concat(P1H1T2e1)

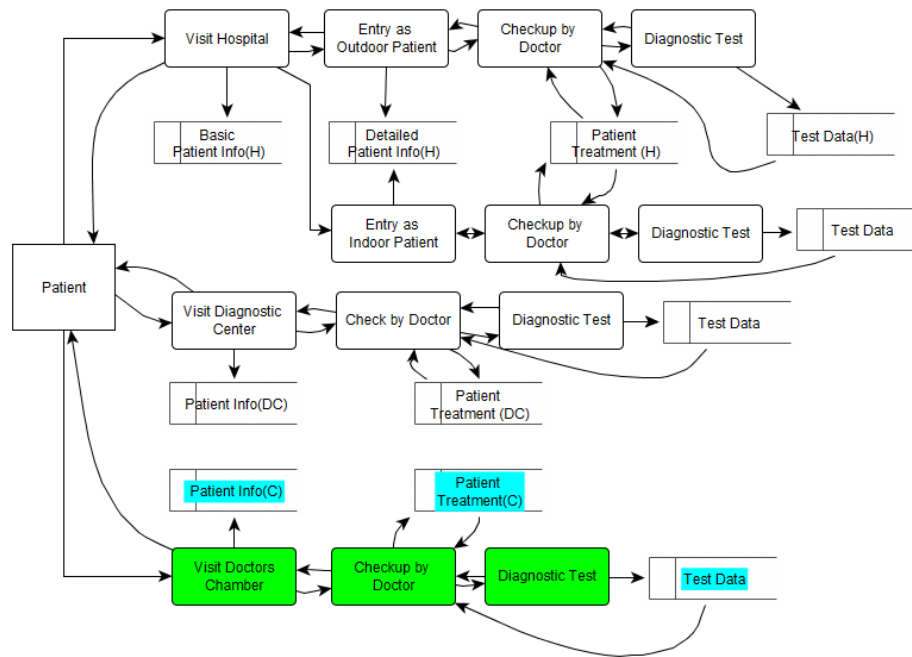


Figure 2.8: Patient treatment in a doctors private chamber

... ..

Case i: Patient P1 visits hospital Hk at timestamp Ti for event e1 with ID concat(P1HkTie1)

... ..

Case n: Patient P1 visits hospital Hk at time stamp Tn for event e1 with ID concat(P1HkTne1)

Now the question is how many possible records that can be evolved in the lifetime of a patient?

Let RL is the total health records for the lifetime of a single patient.

So $RL \subseteq H \times T \times e$

We can estimate an upper limit of RL as follows:

Let life span of a person = y years

Average visit to any health care facility per month= v

Visit per year = 12v

Total health care visits in life span of a person T= 12vy

Average Life expectancy in Bangladesh: Male-70years and Female-72years [194].

If we consider y=71 and v=3/month

So T= 12 X 3 X 71=2556 times

If one visit creates one record, RL=2556

In Bangladesh perspective, health records of a person are stored either in electronic form or

hard copy format and 2556 different records of the same person are stored with 2556 different identities. These records are highly diverse in terms of time (e.g., doing pathological tests in different times), format (e.g., MS Excel or Oracle), and locations (e.g., different hospitals).

For extraction of fruitful knowledge from health data, it is the first requirement to accumulate health records from diverse sources. Privacy of patients needs to be preserved. Record linkage problem is to find an optimum reliable mapping of each patient to his/her health record throughout the patient's lifespan when a national identification number is absent in the patient records.

2.6 Summary

In this chapter, we have presented the background materials to understand the elements and techniques used in record linkage. We have also discussed the challenges faced by the record linkage applications when privacy preservation need to be considered. As our research focus is on health data, the privacy and security issues related to healthcare data and health data servers are also analyzed. Finally, the constraints for record linkage and health data warehousing in Bangladesh and other developing countries are also discussed.

Chapter 3

Missing Data Imputation

In data analytics, missing data is a factor that degrades performance. Incorrect imputation of missing values could lead to a wrong prediction. Missing data create problems in many other application areas such as record linkage or entity resolution. Similarity of records is used for record linkage. Similarity can not be calculated correctly when an attribute's value is missing for some tuples. In this chapter, we propose a new missing data imputation technique, namely SICE, which is a hybrid approach of single and multiple imputation techniques. We also implement twelve popular algorithms to impute binary, ordinal, and numeric missing values and compared the performance of SICE with the algorithms.

This chapter is organized as follows. We provide a brief introduction of missing data problem in Section 3.1. We review related research in Section 3.2. Then in Section 3.3, we present two variations of our proposed method SICE for imputing numeric and categorical data. We compare the results of SICE with some existing methods using local and open-source datasets, which is presented in Section 3.4 and 3.5. In Section 3.6, we discuss the results and limitations of our proposed algorithm briefly. Finally, Section 3.7 presents a summary of the chapter.

3.1 Introduction

In the past few years, the generation of digital data has been increased swiftly, along with the rapid development of computational power. These enable the way to extract novel insights from massive datasets, refereed to big data. In different disciplines such as healthcare, bank-

ing, e-commerce, and finance, data analysts are working to discover hidden knowledge from a vast volume of data [118], [178]. Quality of data is a significant concern to them for fruitful data analytics. Although the output quality of data analysis tasks depends on several factors such as attribute selection, algorithm selection, sampling techniques, etc., a key dependency relies upon efficient handling of missing values [30], [60].

Different machine learning and data mining algorithms are widely used to predict outcomes from large datasets. These algorithms usually make proper prediction unless the data used for training the algorithms are flawed. An essential step of the data analysis and mining process is the refinement of the data on which the system will be trained. This part of the data mining process is called data preprocessing, which is recognized as the most challenging part by the data science researchers [150], [202]. In many cases, data is either missing or incorrectly entered by a human, which results in wrong predictions. Especially, real health-care data are often found very noisy and incomplete, which makes the knowledge discovery task very difficult. One of the main issues regarding the quality of data is missing values. Missing values in a dataset may significantly increase computational cost, skew the outcome, and frustrate researchers [65].

It is a simple solution to ignore the observation with missing values. Usually, no significant problem occurs when there are very few observations with missing values. However, deleting a large number of observations causes a significant loss of information [204]. It also decreases the statistical power and efficiency of the data [115]. Reliable imputation techniques are necessary to solve this issue. Imputation of missing data can help to maintain the completeness in a dataset, which is very important in small scale data mining projects as well as big data analytics.

There are some widely used statistical approaches to deal with missing values of a dataset, such as replace by attribute mean, median, or mode. Many researchers also proposed various solutions targeting the imputation of binary, nominal, or numeric data. In this chapter, we present a new technique for missing data imputation named Single Center Imputation from Multiple Chained Equation(SICE) which is a hybrid approach of single and multiple imputation methods. In summary, we have the following contributions:

- We propose two extensions of popular *Multivariate Imputation by Chained Equation*

(*MICE*) algorithm, namely SICE-Categorical and SICE-Numeric for the imputations of categorical and numeric data.

- Our proposed algorithm SICE adopts the simplicity of single imputation methods and uncertainty of multiple imputation methods.
- We implement twelve existing algorithms to impute binary, ordinal, and numeric missing-values of local health datasets as well as three reliable open-source datasets.
- We compare the performance of our proposed algorithm with existing algorithms and found that our proposed algorithm achieves higher Accuracy, F-measure, and less error than its competitors for imputing binary and numeric data.

3.2 Background and Related Works

In this section, we have presented the necessary background and literature related to missing data imputation. First, we briefly describe the types of missing data. Then we have presented the literature review in two categories: single imputation and multiple imputation.

Typically missing data can be of three types:

- Missing Completely at Random (MCAR): Data are missing independently of both observed and unobserved data. For example, in a student survey, if we get 5% responses missing randomly, it is MCAR.
- Missing at Random (MAR): Given the observed data, data are missing independently of unobserved data. For example, if we get 10% responses missing for the male students' survey and 5% missing for the female students' survey, then it is MAR.
- Missing Not at Random (MNAR): Missing observations are related to values of unobserved data itself. For example, if lower the CGPA of a student, the higher the missing rate of survey response, then it is MNAR.

3.2.1 Single Imputation

Single imputation techniques generate a specific value in a dataset where the real value is missing. This technique requires less computational cost. There are many types of single imputation methods proposed by the researchers. The general procedure is to pick the highest possible response by analyzing other responses. The value may be obtained by mean, median, mode of the available values of that variable. Other approaches, such as machine learning-based techniques, may also be used for single imputation. An illustrative example of how single imputation works is presented below.

In Table 3.1, we can see that there are two missing values in the "Income" column for serial number 2, and 5 which are represented by NA. We can run mean imputation to impute the missing values. Here, for each missing value, only one value will be imputed by the algorithm. Now we will calculate the mean of the available values of the "Income" column.

$$\text{Mean} = (100+100+300+200+200)/5 = 180$$

Table 3.1: A dataset with missing values

Serial	Gender	Income
1	Female	100
2	Female	NA
3	Male	100
4	Female	300
5	Male	NA
6	Male	200
7	Female	200

At this point, the missing values of serial 2 and 5 will be replaced by the mean value of this column, which is 180. Table 3.2 represents the situation after the imputations of missing values. If there are a lot of missing data in a column, and these data are replaced by the same value, the statistical result like standard deviation, variance goes down. In single imputation, imputed values are considered as actual values. Single imputation ignores the fact that the

actual value cannot be predicted for sure by any imputation method. Single imputation based methods do not consider the uncertainty of the imputed values. Instead, they recognize the imputed values as actual values in subsequent analysis. However, these values may have standard errors. These causes bias in the result [9], [80].

Table 3.2: Imputing missing values using single imputation method

Serial	Gender	Income
1	Female	100
2	Female	180
3	Male	100
4	Female	300
5	Male	180
6	Male	200
7	Female	200

In Table 3.3, we can see, there are some missing values in the dataset. If we use a single imputation strategy, we may take "Mode" (most frequent value) of our target column "Death Reason" to fill these missing values. In this example, the mode is "Cancer," so all the missing data will be replaced by "Cancer." However, if we consider the age column, then we can see that the missing values are for the *senior* patients who are more likely to die in Covid19. So, if we just fill all the missing values using only single imputation, it may not correctly address the uncertainty of the dataset and likely to produce bias imputation.

The followings are some prominent research of single imputation based missing data imputation techniques. Grzymala-Busse, J. W., and Grzymala-Busse, W. J. [68] presented a review of existing missing data handling methods in the handbook *Handling Missing Attribute Values*. They have categorized existing methods into sequential imputation and parallel imputation methods and discussed the popular sequential imputations, e.g., case deletion, assigning the most common value, concept-restricted assignment of values. A few parallel imputations were also discussed in their paper, e.g., rule induction, lower and upper approx-

Table 3.3: Analysis of bias for single imputation method

Serial	Age	Death Reason
1	60	Covid19
2	64	NA
3	42	Heart Attack
4	67	Covid19
5	80	NA
6	32	Cancer
7	35	Cancer
8	45	Cancer
9	88	NA
10	33	Heart Attack

imation, attribute value pairing.

In [140], the authors stated the influences and risks of missing data imputation on medical data and how they impact the classification accuracy. The authors compared three averaging methods of data imputations: global average, cluster average, and class average. The importance of using classification techniques after imputation with an algorithm is also discussed in the paper.

Rahman M. [151] presented an imputation technique for missing healthcare data based on a machine learning approach. Here, the author used an algorithm, namely the Fuzzy Unordered Rule Induction Algorithm(FURIA). FURIA is an advancement of a learner algorithm called RIPPER [85]. FURIA produces a few if-then rules depending on the dataset. Later these if-then rules can be used to impute the missing values. The author compared the performance of FURIA with kNN, J48, SVM, and Mean imputation, to impute missing data and found FURIA to be better in terms of sensitivity. Accuracy of FURIA was not always promising than its competitors.

Schmitt P., Mandel J., and Guedj M. selected six of the most popular methods for miss-

ing data imputation from Google search engine and compared the methods using open-access datasets, i.e., iris, e.coli, and breast cancer [168]. They evaluated the effectiveness of these methods using root mean square error (RMSE), Unsupervised Clustering Error, and Supervised Clustering Error. the authors show Bayesian Principal Component Analysis(bPCA) and Fuzzy K-Means(FKM) outperform the other methods.

Amiri M. and Jensen R. [7] presented a missing data imputation technique using Fuzzy-Rough Methods. The paper helps its readers to grasp the concepts of fuzzy-rough sets along with different versions of fuzzy inference and their implementation. The paper used "KEEL," an open-source software, as well as a library that can be used to perform advanced data-mining techniques over a dataset [177]. KEEL has the implementation of algorithms like Fuzzy-Rough Nearest Neighbor (FRNN), which is a classification algorithm. The authors considered FRNN and proposed three missing value imputation methods- Fuzzy-Rough Nearest Neighbors Imputation(FRNNI), Vaguely Quantified Rough Sets(VQRS), and Ordered Weighted Average Based Rough Sets(OWABRS). In the end, FRNNI was found to be performing best among the three proposed algorithms.

In [90], the authors compared seven imputation methods for numeric data. The algorithms are mean imputation, median imputation, predictive mean matching, kNN, Bayesian Linear Regression (norm), Linear Regression, non-Bayesian (norm.nob), and random sample. They used five numeric datasets from the UCI machine learning repository and found that kNN imputation outperformed all other methods.

Support Vector Machine (SVM) is a popular classification algorithm that is widely used for missing data imputation [82], [145]. For a labeled training sample, SVM tries to find an optimal separating hyperplane such that the distance from the hyperplane to the nearest data points is maximized [23]. The larger this distance (i.e., "margin"), the lower the generalization error of the classifier. The classifier is referred to as the maximum margin classifier. The data points that are nearest to the hyperplane are called the support vectors. Several kernel functions have been used in SVM to reduce the computational cost for classification such as the linear kernel, Laplacian kernel, and polynomial kernel.

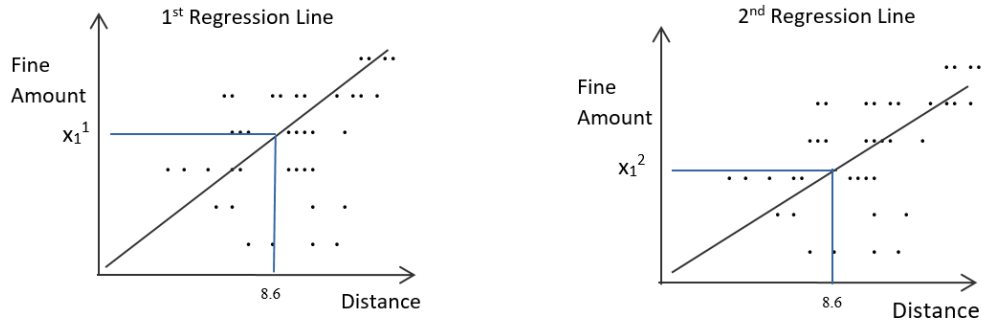


Figure 3.1: Regression lines from two sets of random 100 data taken from 1000 library fine data

3.2.2 Multiple Imputation

Multiple imputation methods produce multiple values for the imputation of a single missing value using different simulation models. These methods introduce the variability of imputed data to find a range of plausible responses. Multiple imputation methods are complex in nature, but they do not suffer from bias values like single imputation. MICE algorithm, proposed by V. S. Buuren and K. Groothuis-Oudshoorn, is widely used for multiple imputation [32]. The working principle of multiple imputation techniques is illustrated next with an example.

In multiple imputation, each missing data are replaced with m number of values obtained from m iterations (where $m > 1$ and m normally lies between 3 to 10). Let us have a dataset of 1000 peoples (shown in Table 3.4) about their distance from a particular library and the amount of late fine the library has imposed on them. The dataset has some missing values in the *fine amount* column. We want to impute the missing values using multiple imputation techniques where the value of m is 10. In each iteration, we will run regression between "Distance from library" and "Fine Amount" by taking 100 random values. In the first imputation, we get x_i^1 for missing values (replacement of the i th missing value of target variable x with first regression). Similarly, in the second imputation, we take another 100 random values and run regression between "Distance from library" and "Fine Amount." Then we fill the i_{th} missing value with x_i^2 (replacement of i_{th} missing value of target variable x with second regression). We will perform these steps ten times to get ten imputations for all missing values of the target variable. Figure 3.1 is an illustration of two imputations using two regression lines. Table 3.5 represents the results of 3 imputations.

Table 3.4: Example of 1000 library fine data with missing values

Serial	Distance from library	Fine Amount
1	1.7 mi	\$11
2	2.1 mi	\$10
3	8.6 mi	NA
4	0.2 mi	\$3
5	6.1 mi	NA
...
...
...
1000	5.3 mi	\$10

Table 3.5: Multiple imputation for table 3.4

Serial	Distance from library	Fine Amount [1st Imputation]	Fine Amount [2nd Imputation]	Fine Amount [3rd Imputation]
1	1.7 mi	\$11	\$11	\$11
2	2.1 mi	\$10	\$10	\$10
3	8.6 mi	\$17	\$16	\$18
4	0.2 mi	\$3	\$3	\$3
5	6.1 mi	\$15	\$15	\$16
...
...
...
1000	5.3 mi	\$10	\$10	\$10

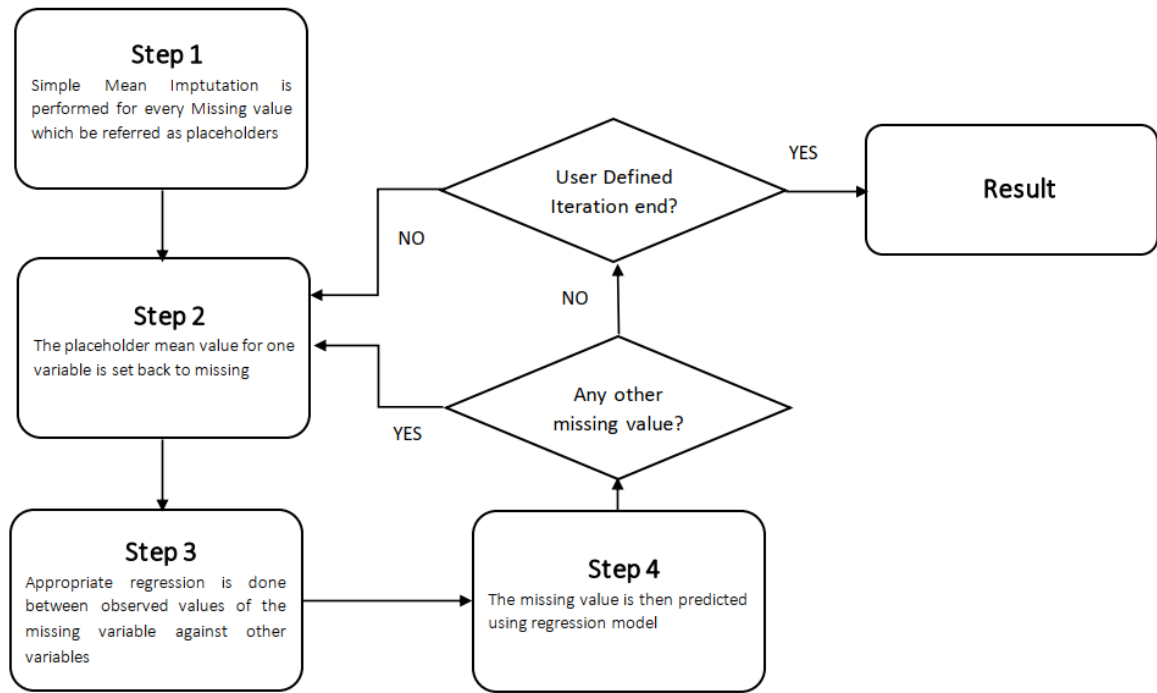


Figure 3.2: Flowchart of MICE

Multivariate Imputation by Chained Equation (MICE) package of "R" is the implementation of the popular MICE algorithm. MICE assumes that data are missing at random (MAR). It pretends the probability of a missing variable depends on the observed data. MICE provides multiple values in the place of one missing value by creating a series of regression (or other suitable) models, depending on its 'method' parameter. In MICE, each missing variable is treated as a dependent variable, and other data in the record are treated as an independent variable. The process is presented in Figure 3.2.

At first, MICE predict missing data using the existing data of other variables. Then it replaces missing values using the predicted values and creates a dataset called *imputed dataset*. By iteration, it creates multiple imputed datasets. Each dataset is then analyzed using standard statistical analysis techniques, and multiple analysis results are provided. As popular single imputation methods, e.g., mean, class-mean, are likely to produce a biased imputation, multiple imputation methods could provide better results.

In the MICE package of R, there are more than twenty methods that can be set for the imputation of missing data [32]. Some methods can be applied only to binary data, and some others work for numeric data. Few methods can be used for any attribute types. Selected methods from the MICE package are discussed below.

3.2.2.1 Predictive Mean Matching

Predictive Mean Matching (PMM) is a general-purpose method for missing data imputation [189]. One advantage of PMM is that imputations are confined to the observed values. PMM can preserve non-linear relations also when the structural part of the imputation model is incorrect. Let, k is a variable with some missing values, and variable l , with no missing data, is used to impute k . The algorithm works in the following way:

1. For non-missing data, linear regression of k on l is done, which produces b (a set of coefficients).
2. A random draw from the posterior predictive distribution of b is made, which produces a new set of coefficients b^* .
3. By using b^* , predicted values for k are generated for all cases.
4. For the cases with missing k , a set of cases are identified that contained observed k whose predicted values are close to the predicted value with missing data.
5. From those close cases, a value is chosen randomly to replace the missing value.
6. Steps 2 to 5 are repeated for every completed dataset.

3.2.2.2 Logistic Regression

Logistic Regression (LOGREG) [197], a popular statistical tool used to analyze a dataset for an outcome where there are one or more independent variables. In logistic regression, the dependent variable is binary. Examples of such data could be YES or NO. Logistic regression generates the coefficients to predict a logit transformation of the probability of presence of the characteristic of output:

$\text{logit}(y) = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + \dots + b_kX_k$ where y is the probability of the presence of the characteristic of output.

3.2.2.3 Polytomous Logistic Regression

Polytomous Logistic Regression (POLYREG) [22] method defines how multinomial target variable Q depends on a set of independent variables, P_1, P_2, \dots, P_m . This is also a generalized

linear model where the random component assumes that the distribution of the dependent variable is Polynominal (n, π) , where π is a vector with probabilities of "success" for each category.

3.2.2.4 Linear Discriminant Analysis

Linear Discriminant Analysis(LDA) [13] calculate posterior probabilities for all incomplete cases and pick imputations, subsequently, from their posteriors. Steps for linear discriminant analysis is given below

1. Calculate the d-dimensional mean vectors from dataset for different classes
2. Calculate scatter matrices
3. Compute eigenvectors (e_1, e_2, \dots, e_d) and their associated eigenvalues $(\lambda_1, \lambda_2, \dots, \lambda_d)$ for the scatter matrices
4. Sort eigenvectors according to the decreasing eigenvalues and choose k eigenvectors with the highest eigenvalues to form a matrix W with $d \times k$ dimension
5. Use W to transform the samples onto new subspace. This can be summarized by the matrix multiplication: $Y = X \times W$

3.2.2.5 Classification and Regression Tree

Classification and Regression Tree (CART) [117] first examines all explanatory variables and determine which binary division of a single explanatory variable best reduces deviance in the response variable. CART and other decision tree-based algorithms have the following key elements:

- Rules to split data at a node based on the value of one variable
- Stopping rules to decide the terminal branch with no more split
- A prediction in each leaf node for the target variable

3.2.2.6 Bayesian Linear Regression

Bayesian Linear Regression (BLR) [35] is a popular statistical method. It is an approach to linear regression, where statistical analysis was undertaken within the context of Bayesian inference. Here linear regression is formed with the help of probability distributions instead of point estimates. Y , the response, is not assessed as a single value, but y is assumed to be drawn from a probability distribution. BLR aims to find out the posterior distribution for the model parameters rather than finding a single best value.

3.2.2.7 Amelia

Amelia, a multiple imputation method, is not included in the MICE package, and a separate R package is available for it. To impute missing values in a specific dataset, Amelia uses a bootstrapping and expectation-maximization algorithm. It creates multiple imputations by multiple iterations [81]. This is helpful since later imputations can be compared to discover trends or to find better results.

3.2.3 Summary of Literature Review and Research Gap

Single imputation based approaches are computationally efficient but may significantly suffer from bias as they do not consider the uncertainty of the missing data. On the contrary, multiple imputation based approaches avoid bias and add uncertainty at the cost of high computational cost. In this era of big data, where a massive volume of data is the typical case for practical datasets, multiple imputation based approaches are challenging to implement. Considering the limitations of both single and multiple imputation based approaches, we are proposing an approach that combines the goodness of both the approaches: simplicity and uncertainty. Our proposed technique for imputation is presented in the next section.

3.3 Proposed algorithm

Multiple imputation based approach such as MICE is a better strategy for handling missing data than single imputation as multiple imputations consider the uncertainty of missing data. As multiple imputation strategy generates m values for a single missing data (where m is a

user-defined number, usually set to 3 to 10), it is complex to use MICE in practical cases with a massive dataset. As the data analyst has to preserve and analyze multiple datasets instead of one. In this section, we propose an algorithm Single Center Imputation from Multiple Chained Equation(SICE). It is an extension of the existing MICE algorithm. We have proposed two variants of SICE, namely SICE-Categorical and SICE-Numeric. Following Algorithm 1: SICE-Categorical imputes missing values of categorical attributes such as binary or ordinal attributes. For better understanding, we also present a flowchart of the SICE, which is applicable for both categorical and numeric version in Figure 3.3. It executes the MICE algorithm for user-defined m times and adds the results in an array. Then a missing value is replaced with the most frequent item of the array.

Algorithm 3.1: SICE-Categorical

Input: x : instances with missing categorical data in a dataset;
 y : instances with no missing data in the same dataset.;
 m : number of imputation defined by user

Output: x' : updated x with imputed missing data

- 1 **for** *each missing value in x* **do**
- 2 | Use MICE to find the the missing value ;
- 3 **end**
- 4 Repeat for m times;
- 5 $miceResult [i] \leftarrow$ imputed data for i_{th} missing value;
- 6 **for** *each row in $miceResult$* **do**
- 7 | SICEresult \leftarrow Mode($miceResult[i,1:m]$);
- 8 | $x' \leftarrow$ x updated with SICEresult
- 9 **end**

The Algorithm 2: SICE-Numeric imputes missing values for numeric attributes. It executes MICE algorithm for a user defined m times and adds the results of each iteration in an array. Then each missing value is replaced by the mean of its corresponding imputed value from the array.

Algorithm 3.2: SICE-Numeric

Input: x : instances with missing numeric data in a dataset;
 y : instances with no missing data in the same dataset.;
 m : number of imputation defined by user

Output: x' : updated x with imputed missing data

```

1 for each missing value in x do
2   | Use MICE to find the the missing value ;
3 end
4 Repeat for  $n$  times;
5  $miceResult[i] \leftarrow$  imputed data for  $i_{th}$  missing value;
6 for each row in miceResult do
7   |  $SICEresult \leftarrow Mean(miceResult[i,1:m]);$ 
8   |  $x' \leftarrow x$  updated with  $SICEresult$ 
9 end

```

3.4 Experimental Design

The block diagram of our imputation and evaluation system is presented in Figure 3.4. At first, a dataset with no missing values is selected as the base dataset. Then, feature selection will be performed, depending on the base dataset, to remove unnecessary attributes. We name this as "Reduced Dataset," which will be used later for performance evaluation of the imputation algorithms. Then we randomly inject 10% missing values to the target attribute of the backup copy of the reduced dataset. After that, we select different imputation algorithms based on the type of the target attribute, i.e., binary or numeric. Then we replace the missing values of the dataset using the selected algorithm of the previous step. In the final step, we evaluate the performance of different imputation algorithms using commonly used matrices such as accuracy, F-measure, or root mean square error.

The algorithms of the MICE package are available in the R environment [32]. The experiments were performed in R-studio. The implemented algorithms are selected based on the attribute type, such as numeric or binary, because some algorithms can able to impute selected attributes. In each experiment, missing values were injected using the 'ampute'

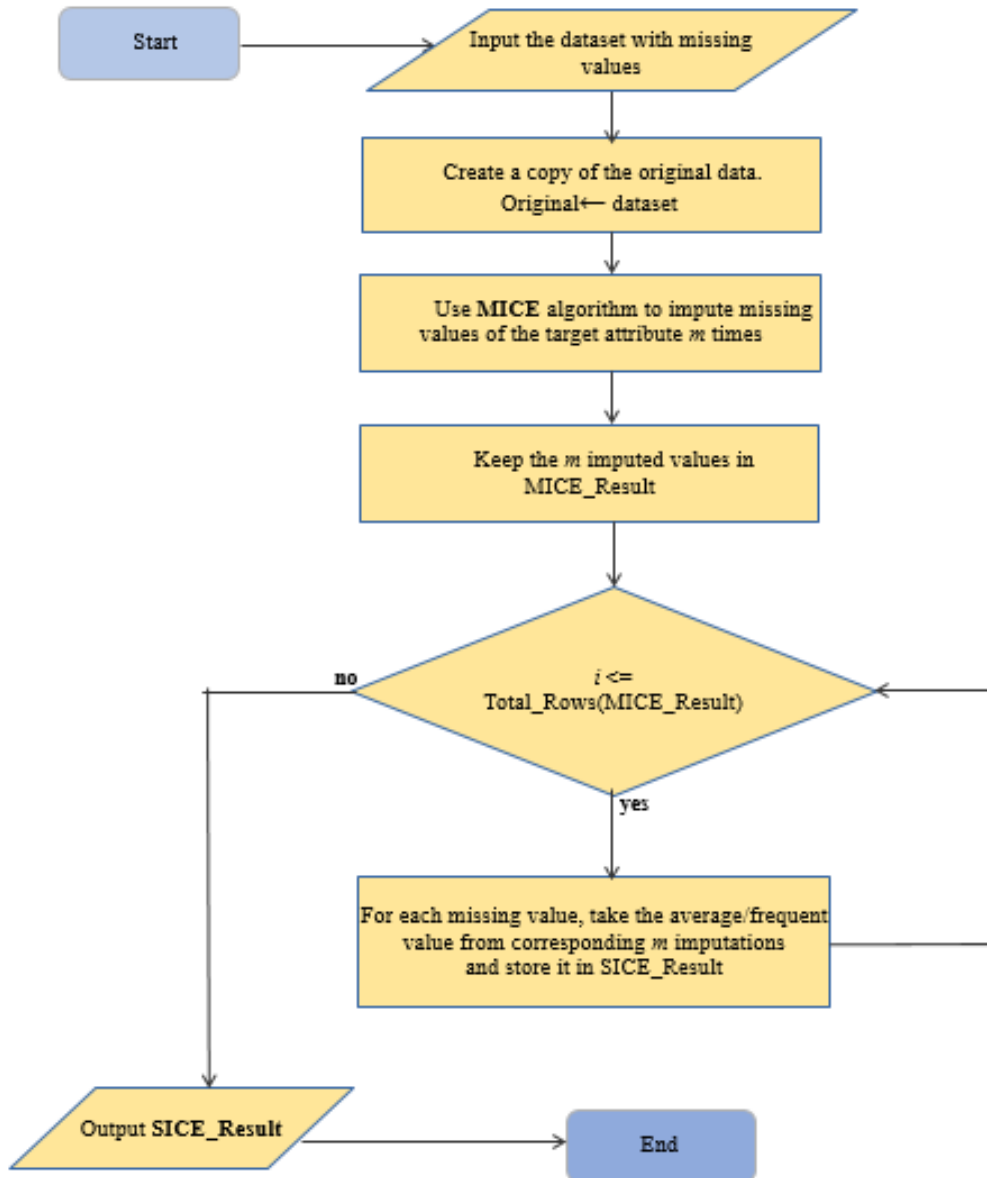


Figure 3.3: Flowchart of SICE

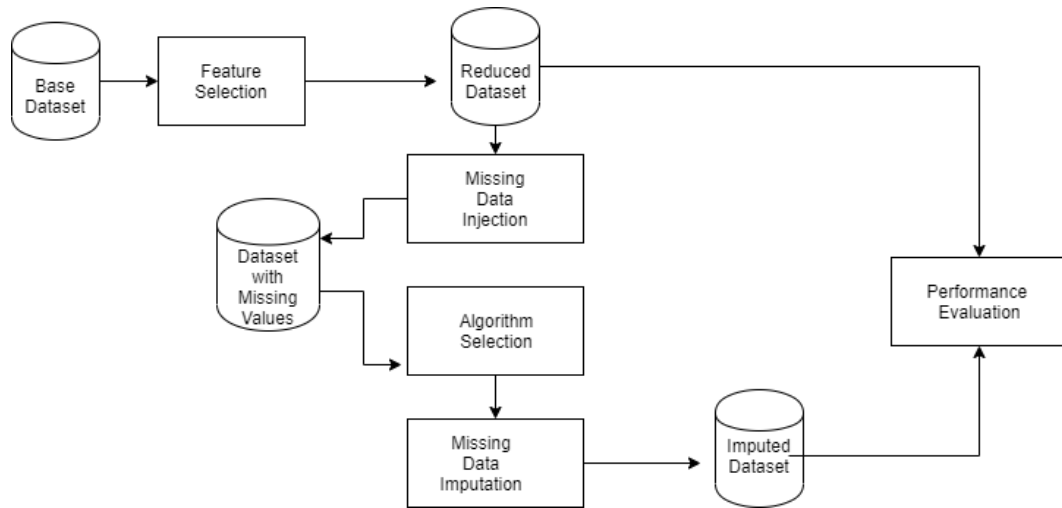


Figure 3.4: Block diagram of the system

function of the MICE package. Later, seven imputations were created using the "mice" function for each of the missing values. The researchers have claimed that to reach a satisfactory efficiency, three to ten number of imputations are sufficient [53]. After trying for different numbers of imputation, we empirically found seven to be a better performer. So we set the value of m (the number of iteration in MICE) to seven. We have run different algorithms by selecting the appropriate method parameter of the MICE function.

3.4.1 Description of the Datasets

We have used four datasets for our experiments. We have collected a local health dataset along with three public datasets from UCI Machine Learning Repository, the Dept. of Mathematics of ETH Zurich, and the Kaggle website. We will briefly describe the datasets here.

3.4.1.1 Local Health Dataset

We have collected patient records from a renowned healthcare center with proper ethical permission under the MoU with the Dept. of CSE, BUET. The dataset has 65 thousand health records, with 13 attributes containing demographic information and diagnosis data of patients. The attributes of the dataset fall into different categories, such as binary, ordinal, and numeric data types. At first, we performed feature elimination to get rid of unnecessary features or attributes, e.g., invoice number, etc. Then we performed the Chi-Square test on these attributes to discover the goodness of fit between them. After that, four attributes

were found as significant. Among them, one was binary, two were nominal, and one was a numeric data type. For our experiments, we have injected 10% missing values in the dataset as per the guidelines of [125].

3.4.1.2 Hair Eye Color Dataset

The second dataset that we have used is a publicly accessible dataset named HairEyeColor [173], which is a distribution of hair and eye color and gender in 592 statistics students. This dataset can be downloaded from the website of the Dept. of Mathematics of ETH Zurich. It is available in R-studio and can be accessed without the use of any external library. We have used the dataset to compare the performance of the binary imputation algorithms. Further detail of the dataset is placed in Section 3.5.1.

3.4.1.3 UCI Car Dataset

We have collected another public dataset to test the performance of the algorithms for imputing ordinal values from the UCI Machine Learning Repository of the University of California, Irvine. The dataset is available at [179]. The number of rows in the dataset is 1728, and the number of columns is 7. Basic statistical descriptions of the target attribute and the results using this dataset are described in Section 3.5.2.

3.4.1.4 Kaggle House Price Dataset

We have taken the last dataset from Kaggle, a popular online community of data scientists and machine learning practitioners. The dataset can be downloaded from [83]. It has 21614 rows and ten columns. Additional information regarding the target attribute and results is placed in Section 3.5.3.

3.5 Results

We have implemented twelve algorithms to compare the performance of SICE. Among them, eight algorithms are included in the MICE package by default, three algorithms are available in different packages of R, and one algorithm (FURIA) is implemented using Weka. The list

Table 3.6: List of existing algorithms implemented for comparison

	Attribute Type		
	Binary	Ordinal	Numeric
Implemented Algorithms	Logistic Regression	Polytomous Logistic Regression (POLYREG)	Amelia
	Predictive Mean Matching (PMM)	Predictive Mean Matching (PMM)	k Nearest Neighbors (kNN)
	Fuzzy Unordered Rule Induction Algorithm (FURIA)	Linear Discriminant Analysis (LDA)	Classification and Regression Tree (CART)
	Support Vector Machine (SVM)	Classification and Regression Tree (CART)	Bayesian Linear Regression (BLR)

of the implemented algorithms for different attributes is presented in Table 3.6. Since each multiple imputation algorithm created seven predictions for each missing value, each algorithm provided seven different datasets as output. However, the result, we have mentioned for each multiple imputation based algorithm, are the best ones from its seven imputations. To evaluate the prediction quality in binary and ordinal attributes, we used Balanced Accuracy, Precision, Sensitivity, Specificity, and F-measure [28]. These properties were calculated and compared using the ‘confusionmatrix’ method from the ‘caret’ [112] package in R. For evaluating the performance of the algorithms on numeric attributes, We used root mean square error (RMSE) which is explained further in Section 5.3.

3.5.1 Performance Comparison for Binary Attributes

Binary attributes are the attributes with two states only. An example of a binary attribute is *gender* when it has only two states: "Male" or "Female." For binary attribute imputation, we have implemented predictive mean matching (PMM), logistic regression (LOGREG), Support Vector Machine (SVM), and Fuzzy Unordered Rule Induction Algorithm (FURIA).

We targeted the ‘gender’ attribute of our local health dataset for imputation as it was the only binary attribute of the dataset. The attribute has 30549 female records and 34451 male records. 10% of total data were injected with missing values as discussed in Section 3.4. Logistic Regression and Predictive Mean Matching were implemented in R-studio using the

MICE package and FURIA was implemented in WEKA. Later, to verify SICEs performance on binary attributes, we tested MICE and SICE on another publicly accessible dataset named HairEyeColor. More information about the dataset is presented in Section 4.1.2. We converted the "Age" attribute of our local health dataset later to binary attribute by using the following rule: Age<18 "Minor" , Age>=18 "Adult". So total tested datasets and the target attributes for imputations are presented in Table 3.7.

Table 3.7: Datasets used for imputation of binary attribute

Dataset Name	Targeted Attribute Name
HairEyeColor	Gender
Local Health Dataset	Gender
Local Health Dataset	Age (Binary)

We implemented MICE and SICE-Categorical using different methods such as PMM, LOGREG, etc. and found that for the binary attribute, SICE-Categorical performs better using the PMM method. The results are presented in Table 3.8. We can see that accuracy and F-measure of SICE is better than MICE, FURIA, and SVM. From Table 3.8, we can see that the F-measure of SICE is 0.656, whereas its closest competitor MICE's F-measure, is 0.546. An illustration of the accuracy and F-measure of the algorithms are presented in Figure 3.5.

The comparison of SICE with MICE for other datasets is shown in Figure 3.6.

Table 3.8: Results for binary dataset "gender"

Algorithm	Accuracy	Sensitivity	Precision	Specificity	F-measure
MICE(PMM)	0.546	0.546	0.546	0.547	0.546
FURIA	0.558	0.558	0.597	0.128	0.468
SVM	0.517	0.188	0.522	0.847	0.276
SICE(PMM)	0.576	0.656	0.656	0.499	0.656

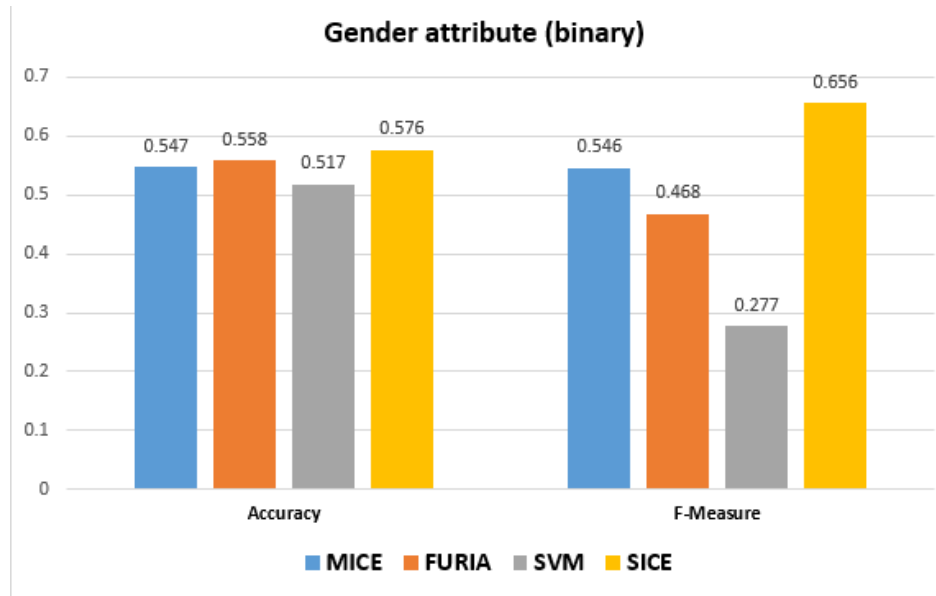


Figure 3.5: Accuracy and F-measure for four algorithms to impute gender attribute

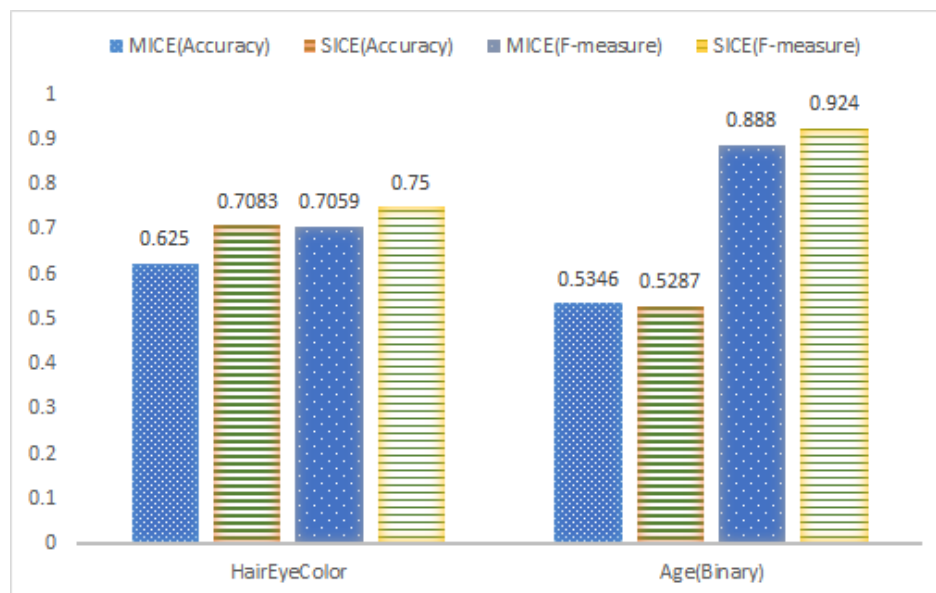


Figure 3.6: Performance comparison of MICE and SICE for additional binary datasets

3.5.2 Performance Comparison for Ordinal Attribute

Ordinal attributes are categorical attributes that have specific levels and maintain order among the levels. For example, if age is converted to categorical data, then that is an ordinal attribute because it has some specific levels with orders, namely- Infant, Child, Adolescent, Adult, and Senior. We used the MICE and RKEEL packages in R for our experiments. We have imputed the Age attribute of our local health dataset described in Section 4.1.1. The variable was initially in a specific date format. First, we converted it into a numeric attribute. We further changed it into ordinal attribute "AgeLevel" by categorizing it into 'Children,' 'Young,' 'Adult,' and 'Senior,' by following the guidelines of [33]. Around 10% of missing data was injected in the target attribute "AgeLevel." Number of rows: 64999, Number of columns: 04, Data in the target (AgeLevel) column: children 4082, Young - 6584, Adult - 44469, Senior 9864

For the imputation of missing data, we implemented four algorithms, PMM, POLYREG, CART, and LDA. The results obtained using MICE and SICE-Categorical are tabulated in Table 3.9. We can see that performance of both MICE and SICE is similar. Figure 3.7 depicted the performance of MICE and SICE-Categorical using PMM and POLYREG, methods for imputing ordinal type of missing data. Both MICE and SICE have shown similar performance with no convincing results. As for ordinal or nominal attributes, there are many choices for a single value; it is difficult to predict the value correctly. However, for a large dataset, the result is expected to improve.

Table 3.9: Performance of MICE and SICE for ordinal attribute using local health dataset

Algorithm	MICE		SICE	
	Accuracy	F-measure	Accuracy	F-measure
PMM	0.503	0.246	0.505	0.238
POLYREG	0.531	0.303	0.532	0.312
CART	0.537	0.318	0.536	0.283
LDA	0.562	0.353	0.561	0.341



Figure 3.7: Performance of MICE and SICE for ordinal data using PMM and POLYREG

We have collected a public dataset to impute ordinal values from the UCI Machine Learning Repository. Details are presented in Section 4.1.3. Some basic statistical descriptions of the target attribute are given below. Number of rows: 1728, Number of columns: 7, Data in the target ("Target") column: 'acc' - 384, 'good' 69, 'unacc' 1210, 'vgood' - 65. The accuracy of MICE and SICE using four methods: PMM, POLYREG, CART, and LDA, are presented in Table 3.10. We can see from the results that our proposed SICE scored the highest accuracy (93.06) and F-measure (81.83) using the CART method as a parameter. The execution time of MICE and SICE in seconds are presented in Figure 3.8. We can see that MICE using the LDA method has the lowest execution time (0.66 seconds), and SICE has slightly higher execution time (0.87 seconds).

Table 3.10: Performance of MICE and SICE for ordinal attribute using UCI car dataset

Algorithm	Accuracy		F-measure	
	MICE	SICE	MICE	SICE
PMM	62.42	74.56	23.41	29.51
POLYREG	83.81	89.59	72.35	76.29
CART	89.01	93.06	76.88	81.83
LDA	80.92	80.92	60.63	64.92

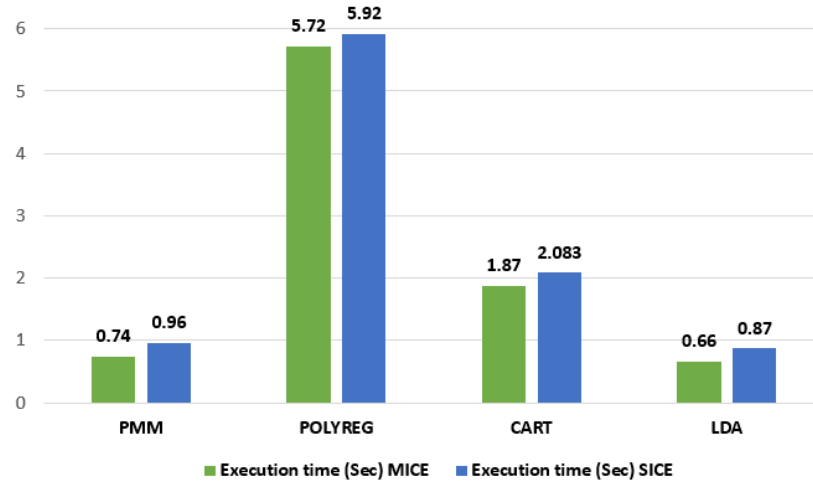


Figure 3.8: Comparison of execution time of MICE and SICE to impute UCI car dataset

3.5.3 Performance Comparison for Numeric Attribute

Numerical attributes are attributes with Numbers. These attributes can be either integer or decimals. An example of the numeric attribute can be the weight of people in kilograms or pounds. We have performed the imputation of a numeric attribute using four different algorithms. They are kNN, Amelia, CART, and BLR. CART and BLR algorithms are included in the MICE package of R. Amelia algorithm has its own package in R named "amelia." The kNN algorithm is available under the "class" package of R.

The experiment was conducted targeting the numeric attribute 'age' in our local health dataset. Age of the people was in years, and other attributes that were present during imputation are one binary attribute and two nominal attributes. To calculate R^2 value we have included one additional attribute "Result" from the raw health dataset. The *Result* attribute has only 2291 values among 65000 rows. So we have reduced the total number of rows for this experiment to 22891.

Number of columns: 05, Our target numeric attribute is AGE column. We have randomly injected 10% missing value. Some useful statistics for the dataset are given below. Min = 1, Max = 95, Range = 1 to 95, Mean = 45.87, Median = 47, Standard Deviation = 18.39, Skewness = -0.18, kurtosis = -0.47. R^2 Value = 0.6470, AIC = 19746.54

The results obtained are tabulated in Table 3.11. To evaluate the algorithms, we calculated and compared the Root Mean Squared Error (RMSE) of each algorithm. The RMSE calculates the absolute fit of the model, and therefore it depicts how closely predicted values

are related to the real values. The lower the RMSE (error value), the better the prediction of an algorithm. To calculate RMSE, we used the following formula:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\text{real_value}_i - \text{predicted_value}_i)^2}$$

It can be seen that our proposed SICE-Numeric using Classification and Regression Tree (CART) method as a parameter gives better results than other investigated algorithms. The prediction error of our proposed SICE is 0.44, which is the lowest compared to its competitors. MICE algorithm using the CART method achieved the second-lowest error, which is 0.67. On the other hand, the MICE algorithms using Bayesian Linear Regression (BLR) method has the lowest execution time, which is 0.23 seconds.

Table 3.11: Performance of the algorithms for numeric attribute of local health dataset

Algorithm	RMSE Score	Execution time
MICE (CART)	0.67	2.03 secs
SICE (CART)	0.44	2.31 secs
MICE (BLR)	1.23	0.23 secs
SICE (BLR)	0.99	0.49 secs
KNN	19.77	0.43 secs
Amelia	25.71	0.58 secs

We have taken the second dataset for numeric imputation from Kaggle, Details of the dataset is presented in Section 4.1.4. The target numeric attribute is price. We have randomly injected 10% missing value (2161 value missing). We converted the price column unit from \$ to k\$. Some useful statistics of the values are given below. Min = 75, Max = 7700, Range = 75 to 7700, Mean = 540.18, Median = 450, Standard Deviation = 367.36, Skewness = 4.02, kurtosis = 34.51, R^2 Value = 0.57, AIC = 296597.4.

We have run MICE and SICE to impute the dataset using CART and BLR methods. We have also run Amelia and kNN algorithms to impute missing values. Price prediction error and execution time are presented in Figure 3.8. We can see that our proposed algorithm SICE using the CART method imputes the dataset with the lowest RMSE error 220, where its close competitor is kNN with RMSE 229. On the other hand, MICE (BLR) has the lowest

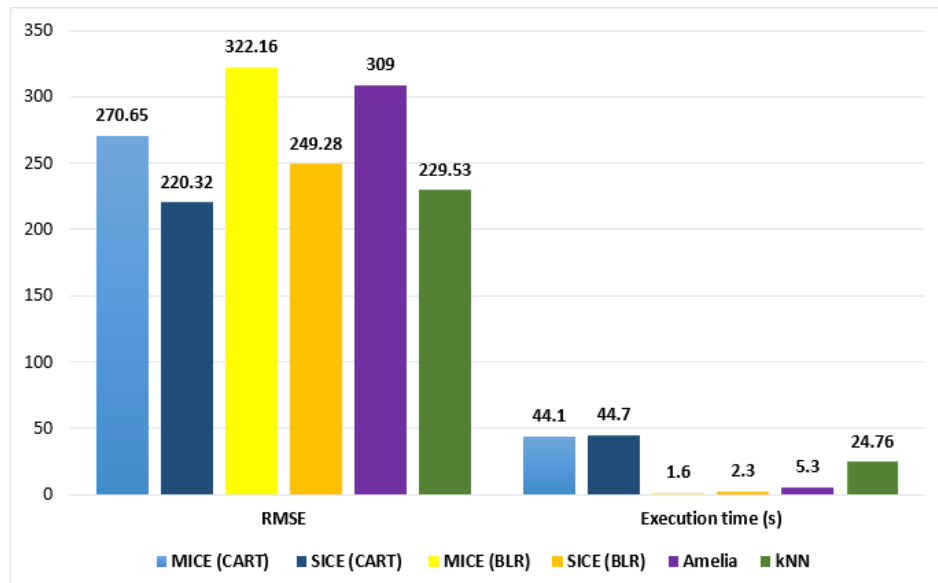


Figure 3.9: Performance of algorithms to predict house prices

execution time 1.6 second, and its close competitor is SICE (BLR) with 2.3 seconds.

3.6 Discussions and Limitation

In this chapter, we have proposed an algorithm Single Center Imputation from Multiple Chained Equation (SICE) with two variants SICE-Categorical and SICE-Numeric to impute missing categorical and numeric data. From the Result section, it can be observed that SICE-Categorical shows better performance over MICE and other implemented algorithms for imputing missing binary data. For all three datasets, SICE achieved 10% to 20% accuracy and F-measure than its competitors. SICE-Numeric also performs better predictions with less RMS error for imputing missing numeric data. That means it provides closer prediction to the correct value than its competitors.

One limitation of SICE-Categorical is that it could not show better performance than MICE for the case of ordinal data. One of the main challenges here is that, for the case of ordinal or nominal data, there may be many states. For example, there are many options for a missing blood group of a person, e.g., B+, B-, O+, O-, AB+, etc. So, it is difficult to impute missing nominal data correctly. In the future, we will focus on improving the SICE-Categorical so that it can perform better for imputing ordinal and nominal data. Another point to notice is that our proposed technique SICE requires slightly higher execution time

than MICE. This is logical as we have extended MICE by adding some additional steps in it. This little increase in the execution time can be overlooked as missing data imputation is performed *offline* in the preprocessing step of a data analytics project.

3.7 Summary

The significance of the imputation of missing data is very high in data analytics. Missing data also adversely affect the performance of record linkage algorithms. Finding a generalized suitable method of missing data imputation for all type of dataset is very challenging. Single imputation based missing data handling methods are comparatively less complex to implement but may provide biased imputations, according to statisticians. On the other hand, multiple imputation based methods consider the uncertainty of a dataset and generate a set of plausible values for each missing data, which are complex to implement. MICE package in R provides the platform to implement Multivariate Imputation of Chained Equations (MICE) technique and support twenty-two methods. In this chapter, we have proposed an algorithm *SICE* for missing data imputation. It is an extension of the popular MICE algorithm. We have presented two variants of SICE: SICE-Categorical and SICE-Numeric to impute binary, ordinal, and numeric data. We have implemented twelve existing methods of missing data imputation and compare their performance with SICE. Experimental results with four different datasets show that our proposed method SICE performed better for the imputation of binary and numeric data. In terms of F-measure, the improvement is around 20%, and in terms of error reduction, the improvement is around 11%. The execution time of SICE is almost equal to MICE. So, we can say that SICE is an excellent choice for missing data imputation, especially for massive datasets where MICE is impractical to use because of its complexity. In the future, we will extend the SICE algorithm for improving its performance further, especially for nominal data.

Chapter 4

Phonetic Encoding for Record Linkage

In this chapter, we explore the second research problem of our thesis. We focus primarily for developing efficient phonetic encoding algorithm that will support record linkage by reducing noise. Phonetic algorithm plays an essential role in many applications including name-matching, database record linkage, spelling correction, noise reduction, search recommendations, etc. Widely used phonetic algorithms such as Soundex and Metaphone are primarily developed for English phonetics. They do not support Bengali Language and show poor performance for Bengali phonetic. Use of Bengali Unicode is increasing in Bangladesh and around the globe with the increasing use of computers everywhere. For example, in different healthcare systems, a patients name can be stored both in English representation of Bengali or Bengali Unicode. Being unable to process Bengali Unicode may lead to failure of linking records within multiple databases. In this chapter, We propose a novel phonetic algorithm nameGist which can efficiently encode Bengali phonetic names in English representation, Bengali Unicode names and English phonetic names.

In Sec. 4.2, we review existing phonetic algorithms for English and Bengali name matching. We discuss the details of our proposed algorithm in Sec. 4.3. Sec. 4.4 describes the datasets that we use to compare the performance of our algorithm nameGist with other algorithms. Sec. 4.5 shows our results and discusses the findings and limitations. We conclude our work in Sec. 4.6.

4.1 Introduction

Phonetic algorithms help to compare and match words by their pronunciation, rather by their spelling. It matches two different names or words with similar pronunciation, by generating similar code. It can be used to quantify the similarity between names by their sound. It has important applications in database indexing, name-matching, spelling suggestion, searching, entity resolution, record linkage, etc. [100], [146].

Record linkage is the process of bringing information together which relates to the same individual from various data sources. It has many applications such as in Healthcare, Finance, Census, etc. Record linkage has much importance in health data integration and analytics because different data source can have the same patients health records. Linkage of health records can have a more significant impact on treatments, improved communication, and health outcomes, etc. By integrating a sequence of health events, it is possible to create an overall health profile of an individual [41]. Misspelling of a patient name in health records can severely affect the integration of health data [104]. To address misspelled names during data integration from different sources, phonetic algorithms can be used.

A phonetic algorithm can withstand an incorrect spelling of a name by generating the same code, which helps to solve the problem of identifying all records of a person during data integration. Phonetic algorithms can support the record linkage process in two ways. First, by reducing noise, they help to improve the accuracy of record linkage algorithms. Second, they have an inherent privacy preservation characteristics that support privacy-preserving record linkage. Soundex, the oldest and most commonly used phonetic algorithm, works well with the English names. There are other several popular phonetic algorithms like NYSIIS, Metaphone, Match Rating Codex, etc. focusing on English phonetics. These algorithms are also not generalized for other languages. None of them supports Bengali phonetic in the English language (we refer it as "Bengali phonetic" in this article) or Bengali phonetic in the Bengali language (we refer it as "Bengali Unicode" in this article). Recently some algorithms have been proposed for Bengali phonetic names [100, 105]. However, these algorithms do not support Bengali Unicode. Performances of these algorithms are also not up to the mark for English phonetic names. If an algorithm performs well with only Bengali phonetic, it will not be able to perform well when a database has a mixture of names from Bengali and

English phonetic. This is a challenge that need to be addressed.

4.1.1 Significance of Bengali Language

Nearly 160 million people live in Bangladesh and speak in Bengali [160]. More than 250 million people speak in Bengali worldwide [161]. It is the national language of Bangladesh and the second most widely spoken language in India. It is also the official state language of West Bengal and Tripura. It is counted as the seventh most spoken native language in the world [121]. The Bengali language also took place in the world history for language movement, excelled on 21st February 1952. Later, United Nations (UN) declared 21st February officially as International Mother Language Day to honor Bengali language movement [162].

Before going for the further details, we are presenting the phonetic transcription of all Bengali vowels and consonants in Roman script following three standards. They are Bangla Academy Romanization (BN), National Library at Kolkata Romanization (NLK), and International Phonetic Alphabet Romanization (IPA). Vowels are presented in Table 4.1 and Consonants are presented in Table 4.2.

4.1.2 A Motivational Example

The Government of Bangladesh has taken various steps to implement Health Information System since 2008. Health data are now stored in electronic format by many healthcare service providers. The Government also started the process of integration of health data to build data warehouse [106], [108]. There is no general Health ID which is used in all hospitals. Few modern hospitals provide ID cards or registration numbers to their patients which are not recognized by other healthcare centers. Misspelling is a typical scenario in Bangladesh, as the literacy rate is low, and lack of awareness is a common phenomenon [104]. People often make mistakes writing their name in the application forms. Moreover, while filling out the form, people may put their name in both Bengali Phonetic and Bengali Unicode. Misspelling of a name can also occur when typing on the computer. Misspelling can cause a severe problem in the integration of health data.

For example, if the health record of a patient Tanveer Rahman is taken from various medical records, we may see the name is written in different formats and also often misspelled.

Table 4.1: Bengali vowels and other miscellaneous characters

Symbol	BA	NLK	IPA
অ	a	a	[ɔ]/[o]
আ	ā	ā	[a]
ই	i	i	[i]
ঈ	ī	ī	[i]
উ	u	u	[u]
ঊ	ū	ū	[u]
ঋ	r	ṛ	[ri]
এ	e	ē	[e]/[ɛ]
ঐ	ai	ai	[oi]
ও	o	ō	[o]
ঔ	au	au	[ou]
ঃ	h	h	varies
ং	ng	m̃	[ŋ]
্	õ	m̃	[~] (nasalization)
্য	y	y	varies
্ব	w/v	v	varies
ক্ষ	kʂ	kʂ	[kʰɔ]
জ্ঞ	jñ	jñ	[gɔ]
শ্র	śr	śr	[ʃɔ]

Table 4.2: Bengali consonants

Symbol	BA	NLK	IPA	Symbol	BA	NLK	IPA
ক	k	k	[kɔ]	ত	t	t	[tɔ]
খ	kh	kh	[kʰɔ]	থ	th	th	[tʰɔ]
গ	g	g	[gɔ]	দ	d	d	[dɔ]
ঘ	gh	gh	[gʰɔ]	ধ	dh	dh	[dʰɔ]
ঙ	ng	ñ	[ŋɔ]/[uɔ̃]	ন	n	n	[nɔ]
চ	c	c	[tʃɔ]	প	p	p	[pɔ]
ছ	ch	ch	[tʃʰɔ]	ফ	ph	ph	[fɔ~pʰɔ]
জ	j	j	[dʒɔ]	ব	b	b	[bɔ]
ঝ	jh	jh	[dʒʰɔ]	ভ	bh	bh	[bʰɔ]
ঞ	ñ	ñ	[nɔ]	ম	m	m	[mɔ]
ট	ṭ	ṭ	[tɔ]	য	y/j	ÿ	[dʒɔ]
ঠ	ṭh	ṭh	[tʰɔ]	য়	ÿ	y	[ɛɔ]/-
ড	ḍ	ḍ	[dɔ]	র	r	r	[rɔ]
ড়	ṛ	ḍ	[rɔ]	ল	l	l	[lɔ]
ত	ḍh	ḍh	[dʰɔ]	শ	ś/sh	ś	[ʃɔ]
ড়	ṛh	ḍh	[rɔ]	ষ	ṣ/sh	ṣ	[ʃɔ]
ণ	ṇ	ṇ	[nɔ]	স	s	s	[sɔ]
Continued to Right-side				হ	h	h	[hɔ]

Table 4.3 shows some possible misspell of a name Tanveer Rahman as Bengali Phonetic (English representation of Bengali name) and তানভীর রহমান as Unicode Bengali. Another problem is, a patient uses different salutations with his name while filling out the form at different times, e.g., Mr., Sree, Advocate, Ms., Dr., etc. This can cause a significant complication and error during the integration process. Therefore it is vital to eliminate noise and correctly match the patients records.

4.1.3 Contributions

In this work, we have the following contributions:

- We propose an algorithm nameGist, which is the only algorithm, to the best of our knowledge, to support Bengali Unicode name matching.
- Our proposed algorithm also performs significantly better than other existing Bengali phonetic algorithms for name matching.

Table 4.3: Example of misspelling

Original Patient Name	Misspelled Name
	Mr. Tanvir Rahman
	Md. Tanovir Rahman
Tanveer Rahman	Tanvir Rohoman
	Mohammad Tanvir Rohman
	Tanver Rohoman
	তানভির রহমান
তানভীর রহমান	তানবির রহমান
	তানভীর রাহমান

- The nameGist can efficiently process English phonetic names (American/British names) and gives competitive results with popular English phonetic algorithms.
- Our algorithm also supports matching the mixture of Bengali Phonetic, Bengali Unicode and English names at the same time, which can solve the record linkage problem as it is more generalized than other algorithms.

4.2 Literature Review

Researchers have proposed many phonetic algorithms over the past century with various motivations. In this section, we have presented a few popular algorithms among them. We have also presented a brief overview of the recently proposed Bengali phonetic algorithms and their limitations.

4.2.1 Widely Used Phonetic Algorithms

Some of the popular Phonetic Algorithms are:

1. Soundex (1918)
2. NYSIIS (1970)

3. Match Rating Codex (1977)
4. Metaphone (1990)

4.2.1.1 Soundex

Soundex is the most commonly used phonetic algorithm [156], which was developed in 1918 to help analyze US census data. Most of the other phonetic algorithms, proposed later, are variations and enhancements of Soundex.

The Soundex algorithm encodes words as a letter (A to Z) followed by three numerical digits, e.g., B123, K251, etc. The grouping of numerical digit is generated by the place of articulation of the different sounds. For example, d and t are given the code number 3, while b, f, p, and v are given the code number 1.

To generate the Soundex code, the first letter is kept, and then all vowel sounds, including w and h, are removed. The output is a letter plus three number code (X####) for an input string. This simplified encoding of English language makes Soundex a powerful tool for comparing words. Although Soundex is the most commonly used phonetic algorithm, it generates the same code for different last names which may create ambiguity in Record Linkage applications. For example, the Soundex algorithm generates the same code I565 for Imran Hossain and Imran Khan.

4.2.1.2 NYSIIS

NYSIIS [52] is a phonetic algorithm developed in 1970 by the New York State Identification and Intelligence System. It transforms a word into a phonetic code. Like Soundex, it is primarily intended for name matching. The algorithm shows higher accuracy while dealing with American names as it was specially developed for that case.

4.2.1.3 Match Rating Codex

Match rating codex [165] was developed by Western Airlines in 1977, and it finds out whether two names are pronounced similarly or not. It has a much simpler encoding rule but a lengthy set of comparison rules. The encoded name is called personal numeric identi-

fier. One of the encoding rules is to reduce codex to 6 letters by joining the last 3 letters with the first 3 letters.

4.2.1.4 Metaphone

Metaphone [147] [148] algorithm was published by Lawrence Phillips in 1990. Its encoding process follows pronunciation rules to produce a more accurate encoding. A new version of this algorithm is also published by the same author, which he named Double Metaphone. This algorithm supports a few other languages than English, e.g., Greek, French, Spanish, etc. In 2009 Metaphone 3 was published, which achieved an accuracy of approximately 99% for English words.

In summary, above phonetic algorithms perform well for English names, e.g., American or British Names. They Show poor performance while dealing with Bengali and similar Indian/ South-Asian phonetic names, represented in English. Additionally, the above algorithms cannot encode Bengali and similar Unicode names correctly. Their performances are discussed elaborately in the "Result" section.

4.2.2 Algorithms for Bengali Phonetic Names

To solve the problem regarding Bengali phonetics, few works have been proposed recently. The notable works are listed below.

4.2.2.1 NameValue and NameSig

NameValue algorithm was proposed by S.I. Khan and A.S.M.L. Hoque for phonetic encoding of names to support record linkage [103], [102]. Later, the authors performed some minor updates to the algorithm, changed its name to NameSig and publish the algorithm in [105]. The algorithm is presented below.

Algorithm 4.1: NameValue

- 1 Remove salutation;
 - 2 Remove vowels unless beginning of the name or succeeding white space;
 - 3 Encode q/Q/k/K to k;
 - 4 Encode j/J/g/G/z/Z to j;
 - 5 Use Code Table to map the characters;
-

Algorithm 4.2: NameSig

Input: Patient name**Output:** NAMEVALUE of the inputted name**Begin:**

- 1 Delete Title and Salutation;
- 2 Delete a/A, e/E, i/I, o/O, u/U unless beginning of name or after white space;
- 3 Delete white space;
- 4 Convert g/G/j/J/z/Z to g;
- 5 Convert k/K/q/Q to k;
- 6 Mask unambiguous and significant characters using Code Table;

End:

A real health dataset with 633609 patient records from different hospitals of Bangladesh were used to test the performance of the algorithm. It was found from the experimental results that NameSig achieved 87% accurate phonetic codes to link patients' records. Further details of how NameValue or NameSig algorithms encode names can be found in Section 5.6.4.

The limitations we found in NameSig algorithm are given below:

1. Character mapping was only considered for English vowel. It does not consider Bengali vowel which is greater in number than English vowel.
2. More similar sounding characters can be mapped but ignored. For example

b/v to b

e/u/y to e etc.

3. For a simple test case Tanveer Rahman and Tanbir Rahman this algorithm fails to generate proper code.
4. Bengali Unicode is not supported, e.g., তানভীর রহমান can not be processed.

4.2.2.2 Modified NameSignificance

A. B. A. Khan et al., proposed a new algorithm named Modified Name Significance [100] which was an updated version of Name Significance Algorithm. Their algorithm introduced a syllable split method to identify misspelled names. It splits the names into syllables according to English representative letters of Bengali consonants and vowels followed by the simplified mapping of individual letters. It has better accuracy for Bengali phonetic names than most of the standard phonetic algorithms. We found the following limitation of this algorithm:

1. There were no proper steps mentioned how syllable split works. After implementing the algorithm we found from various examples that this method was not standard, e.g., tohidul has 2 syllables: [tohid][ul], but the algorithm assumes it as one syllable.
2. If the repetitive character is removed then right syllable may not be found in the next step. E.g.,

Shajjad > shajad > [sha][jad], it should be [shaj][jad]

Hannan > hanan > [hanan], it should be [han][nan]

3. Syllable split method is very poor detecting Bengali vowel mark changes. for example, Tanvir Rahman, Tanovir Rahoman for this input this algorithm fails, while other standard algorithms successfully identify the changes.
4. Bengali Unicode is not supported, e.g., তানভীর রহমান can not be processed.

4.2.2.3 Double Metaphone Encoding Technique

Zaman et al. introduced a Double Metaphone encoding technique in the context of Bengali in [182], that can be used in Bengali name searching and matching. It encapsulates the

complex spelling rules for Bengali. Nothing is mentioned in their paper about how the technique performs against the real world dataset.

4.2.2.4 Bengali Phonetic Encoding

Zaman et al. proposed a phonetic encoding for Bengali in [181] which is based on the Soundex algorithm. According to them, spelling checkers can provide a better suggestion for misspelled names using this encoding. This is also a theoretical approach, and no implementation details are given.

4.2.3 Summary of Literature Review and Research Gap

We can see from the above literature review that there are several popular phonetic algorithms such as Soundex, NYSIIS, Metaphone, Match Rating Codex whose focus is English phonetic names. These algorithms are also not generalized for other languages. None of them supports Bengali Unicode and show poor performance for Bengali Phonetic names. Recently some algorithms have been proposed for Bengali phonetic names which do not support Bengali Unicode. Performances of these algorithms are also not up to the mark for English phonetic names. If an algorithm performs well with only Bengali phonetic, it will not be able to perform well when a database has a mixture of names from Bengali and English phonetic. So a novel algorithm is needed that could perform well for Bengali phonetic, Bengali Unicode and English phonetic names and also for the mixture of all these types. That is why we come out with nameGist which is going to be presented in the next section.

4.3 Proposed Algorithm: nameGist

The algorithm nameGist is an extension of our ongoing research on phonetic encoding to support record linkage. We first proposed the NameValue algorithm for phonetic encoding of Bengali names [103]. Then, with few modification, we named the algorithm as NameSig [105]. Later, it was further improved to Modified NameSignificance [100]. Details of these algorithms are presented in Section 4.2.2. The nameGist has some unique features that were not present in NameValue or NameSig, which is already presented in Section 4.1.3.



Figure 4.1: Vowel marks in phonetic and unicode Bengali name

Our proposed algorithm nameGist can encode- Bengali phonetic names, Bengali Unicode names, English (American/British) names. We have analyzed a health dataset of seventy-one thousand patients and tried to figure out the causes of misspelling names. We found that even in misspelled names, the core letters always remain the same. People often get confused about how to represent the vowel marks. Most of the errors are introduced because there is no direct representation of the Bengali vowel mark in the English alphabet. There are 26 letters in English and 50 in Bengali. So there is more than one representation of a Bengali alphabet to English, which causes more errors. Our algorithm is based on the proper use of the following ideas:

1. Detecting and removing the vowel marks to recover core letters.
2. Mapping similar sounding words to one.

We have illustrated our idea in Figure 4.1 . Figure 4.1 indicates the vowel marks of a single name in Bengali Unicode and Bengali phonetic in English representation. and the Table 4.4 shows the same name with different writing variations. Here the *underlined* symbols are the vowel marks. From the Figure 4.1, we see that if we remove vowel marks we get the core (gist) letters of a name (Without Vowel Marks column), and then by changing the similar sounding letters to one letter, we get unique letters of a name, despite their misspelling.

4.3.1 Vowel Marks

Table 4.5 shows different vowel marks in Bengali and their representations in English. Here Bengali Unicode column has the Unicode Bengali vowel marks, and Bengali Phonetic column contains the equivalent English representations of the vowel mark.

Table 4.4: Conversion of names with vowel marks and spelling variations to the "gist" name

Real Name	Vowel Marks	Without Vowel Marks	Change Similar Letters
tanveer rahman	tanveer rahman	tnvr rhmn	tnbr rhmn
tanbir rahman	tanbir rahman	tnbr rhmn	tnbr rhmn
tanovir rohman	tanovir rohman	tnvr rhmn	tnbr rhmn
তানভির রহমান	ত ান ি ভ র র হ ম ান	তনভর রহমন	tnbr rhmn
তানবীর রহমান	ত ান ব ী র র হ ম ান	তনবর রহমন	tnbr rhmn

Table 4.5: Vowel marks mapping

Bengali Unicode	Bengali Phonetic	Bengali Unicode	Bengali Phonetic
া	a		
ি	i, e	ী	i, e
ু	u	ূ	u
ে	a	ৈ	oi
ৌ	o	ৌ	ou, ow

4.3.2 Similar Sounding Alphabet Mapping

Based on our analysis, we have proposed a mapping of similar sounding letters. The mapping is listed in Table 4.6 for vowels and 4.7 for consonants respectively.

4.3.2.1 Similar Sounding Vowel Mapping

Table 4.6 shows the mapping of similar sounding vowels. The column English Letter has the possible English phonetic representation of the column Bengali vowel which contains all Bengali vowels. The Changed Letter column contains similar sounding alphabets. Most people are using the vowels ambiguously in writing names in both Bengali and English. So we change all vowels to a which minimizes noise and improves the performance of matching.

Table 4.6: Similar sounding vowel mapping

Bengali vowel	English Letter	Changed Letter	Example Names
অ	a, o	a	arnob, ornob
আ	aa	a	abedin
ই	e, i	a	ira, era
ঈ	ee, i	a	-
উ	u	a	umar, omar
ঊ	uu	a	-
ঋ	-	a	-
এ	a	a	ahsan, ehsan
ঐ	-	a	-
ও	o	a	omar, umar
ঔ	-	a	-

4.3.2.2 Similar Sounding Consonant Mapping

For consonant, we map similar sounding alphabets. Table 4.7 shows the mapping of similar sounding consonants. The column English Letter has the possible English phonetic representation of the column Bengali Consonant which contains all Bengali consonants. The Changed Letter column has the similar sounding alphabets. A null means the character(s) will be removed.

4.3.3 Algorithm: nameGist

We are presenting our algorithm below. The steps of our nameGist algorithm for Bengali Phonetic (Bengali in English representation) and Bengali Unicode name illustrated in Figure 4.2 and Figure 4.3 respectively. Steps of the algorithm are also explained in the following texts.

Table 4.7: Similar sounding consonant mapping

Bengali Consonant	English Letter	Changed Letter	Example Names
ক	k	k	kalam
খ	kh	k	bokhtiar, boktiar
গ	g	s	ragib, julhaj, julhas
ঘ	gh	s	raghib
ঙ	-	null	-
চ	c, ch	s	cadni, chadni, chowdhury
ছ	ch, s	s	choudhury, sakib, chakib
জ	j, z	s	majbah, mazbah, mesbah
ঝ	jh, zh	s	Zhahran, Jhahran
ঞ	-	null	-
ট	t	t	topon,tapon
ঠ	th	t	Mitun, mithun
ড	d	d	Dalim
ঢ	dh	d	Dhalim
ণ	n	n	Arnob
ত	t	t	mitun
থ	th	t	mithun
দ	d	d	chowduri
ধ	dh	d	chowdhuri
ন	n	n	noman
প	p	p	polash
ফ	ph, f	p	faisal, foisal
ব	b	b	tanbir
ভ	bh, v	b	tanvir
ম	m	m	imran
য	j, z, g	s	majbah, mazbah, mesbah
র	r	r	rabbani
ল	l	l	laboni
শ	sh	s	rashida, rasida
ষ	sh	s	-
স	sh, s	s	shohag, sohag
হ	h	h	hasan
ড়	r	r	-
ঢ়	r	r	-
য়	a, y	a	kaisar, kaysar
ৎ	t	t	-
ং	-	null	-
ঃ	-	null	-
ঁ	-	null	-
্	-	null	-

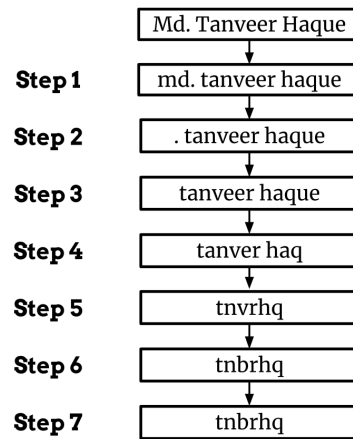


Figure 4.2: Steps of nameGist algorithm for names written in English

Algorithm 4.3: nameGist**Input:** Name-Dataset**Output:** Phonetic Code Dataset**Begin:**

- 1 **for** each name in Name-Dataset **do**
- 2 Convert to lower-case letters (ignore if Unicode Bengali);
- 3 Remove Title and Salutation;
- 4 Remove all unnecessary symbols, extra spaces of start and end;
- 5 Remove double characters and ue if found at the end of a word (ignore if Unicode Bengali);
- 6 Detect and Remove all vowel marks and spaces ();
- 7 Detect and Replace similar sounding Bengali alphabets from the text;
- 8 Remove all double characters;
- 9 Insert the value in the Phonetic Code Dataset;
- 10 **end**

End:

Step 1: Make the name lower-case (ignore for Unicode Bengali name:) In this step, we make all the English characters to lowercase, e.g., Md. Tanveer Haque to md. tanveer haque etc. It improves name processing. This step is ignored for Unicode Bengali names.

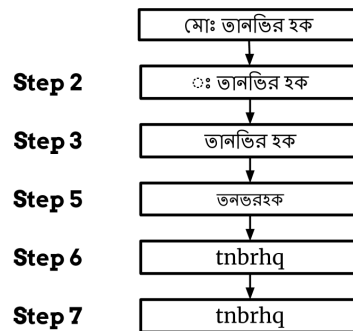


Figure 4.3: Used steps of nameGist algorithm applicable for names written in Bengali

Step 2: Detect and Remove Title and Salutation: Then we detect title and salutations, e.g., Mr., Mrs, Dr., Md., Mohammad, etc. and remove them. These parts are randomly used by the persons most of the time, so we remove it to avoid ambiguity.

Step 3: Remove all unnecessary symbols, start-end extra spaces: It is common that the names include unnecessary symbols like ., , :, -, etc. In this step, we remove those unnecessary symbols and remove all unnecessary start-end spaces.

Step 4: Remove double characters and "ue" if it is found at the end of any word (ignored for Unicode Bengali name:) In this step, we remove all double characters and replace it with one letter. Then we delete the letters ue if it is found at the end of a word. E.g., tanveer haque to tanver haq etc. This step is also ignored for Bengali Unicode names.

Step 5: Detect and Remove all vowel marks and spaces: After the pre-processing is completed, our algorithm intelligently detects all vowel marks and remove them. In this step, all spaces () are also get removed. E.g., tanver haq to tnvrhq etc.

Step 6: Detect and Replace similar sounding Bengali alphabets from the name: In this step, our algorithm detects all similar sounding letters and replace them with a single letter. E.g., tnvrhq to tnbrhq etc.

Step 7: Remove all double characters: Finally, we again check for double characters and remove them if found. Because of steps 5 and 6, in some cases, double characters may be placed together, coincidentally. This step improves the detection rate of our algorithms,

Table 4.8: Sample of Bengali phonetic dataset

Real name	Variation 1	Variation 2	Variation 3	Variation 4
Tanveer Rahman	Tanveer Rahman	Tanver Rahman	Tanvver Rahman	Tanveer Rahaman
Rashid Al Shafee	Rashed Al Shafee	Rashed Al Shafi	Rashid Al Safi	Rashed Al Safi
Mohammed Abdul Kayum Masum	Md Abdul Kayum Masun	Mohammod Abdul Kayum Masum	Mohammad Abdul Kayum Masum	Mohammed Abdul Qaium Masum

which has been found from empirical results. E.g., Tanveer Rahman to `tnvr̄rhmn` then to `tnvr̄hmn` etc.

4.4 Description of the Datasets

We have tested our algorithm using the following datasets, considering four scenarios:

1. Bengali name in English representation (Bengali Phonetic) dataset
2. American (US)/British (UK) name dataset
3. Bengali Unicode name dataset
4. Mixed dataset containing Bengali Phonetic, Bengali Unicode, and English names

4.4.1 Bengali Name in English Representation (Bengali Phonetic) Dataset

In this dataset, we have a total of 292 data. We collected the dataset used in [100]. The dataset contains only Bengali phonetic names as input. In this dataset, different misspelled names lie with the correct one in a row. A snapshot of the dataset is given in Table 4.8.

4.4.2 American/British Name Dataset

To see how our algorithm works with English names, we have used two different English name dataset. We have collected the first dataset from open name database [158]. We down-

Table 4.9: Sample of US/UK English name dataset

Real name	Variation
Jinny Kopper	jinna kopper
Tasha Recchia	tyshy resshiy
Benjamin Vonk	venjymin vonk

Table 4.10: Sample of Bengali unicode dataset

Real name	Variation 1	Variation 2	Variation 3
তানভীর রহমান	তানবীর রহমান	তানবির রহমান	তানবীররহমান
মোহাম্মদ কফিল উদ্দীন	মোঃ কফীল উদ্দীন	কপিল উদ্দিন	মোহাম্মদকপীলউদ্দীন
নাইম মাহমুদ	নাইম মাহমুদ	নইম মাহমুদ	নাইমমাহমুদ

loaded first name and last name from the source and randomly joined the names and then took 2000 names. We then duplicated the names and added 10%-30% random errors to the dataset. We have collected the second dataset from the US Social Security Administration [163] (first names) and US Census Bureau [164] (last names). Then we joined the first and last names randomly and took 2000 full names from the dataset. Finally, we duplicated the names and added random error to the dataset. Both the datasets have 4000 names. A snapshot of the English dataset is given in Table 4.9.

4.4.3 Bengali Unicode Name Dataset

This dataset has a total of 4044 data. The dataset is collected from an online name collection website [159]. This dataset contains only Bengali Unicode names. In this dataset, different misspelled names lie with the correct one in a row. A snapshot of the dataset is given in Table 4.10.

Table 4.11: English- Bengali mix dataset

Real name	Variation 1	Variation 2	Variation 3	Variation 4
Tanveer Rahman	Tanveer Rahman	Tanver Rahman	Tanvver Rahman	তানভীর রহমান
রাশিদ আল শাফি	Rashid Al Shafee	Rashed Al Shafi	Rashid Al Safi	Rashed Al Safi
Mohammed	Md Abdul	মোহাম্মদ আব্দুল	Mohammad	Mohammed
Abdul Kayum	Kayum Masun	কাইয়ুম মাসুম	Abdul Kayum	Abdul Qaium
Masum			Masum	Masum
Felica Mellos	phelicy mellos	-	-	-
Toby Caretto	tovy cyretto	-	-	-

4.4.4 Bengali Phonetic, Bengali Unicode, and US/UK English Mixed Name Dataset

The previous datasets contain either Bengali Phonetic, Bengali Unicode or English names. However, in reality, in a dataset, all the variations may be possible. If we consider the motivational example of Section 4.1.2, in the health data warehouse, both Bengali phonetic and Unicode may be present simultaneously. Again, in Bangladesh, many foreigners live and work. So the warehouse will contain English names. To test our algorithm in this scenario, we combined previous three datasets (Bengali Phonetic name dataset, Bengali Unicode dataset, and first English dataset) and took 4376 names randomly from the combined dataset. A snapshot of the dataset is given in Table 4.11.

4.5 Results and Discussion

In this section, we will present the results of nameGist and other implemented algorithms against the datasets described in Section 4.4. We first discuss the experimental setup and our measurement parameters briefly.

4.5.1 Experimental Setup

We have used the following tools for implementations:

Table 4.12: Description of the confusion matrix

	Input	Expected Output	Actual Output	Verdict
TP	Similar Names	Same Unique Code	Same Unique Code	Correct
FN	Similar Names	Same Unique Code	Different Unique Code	Wrong
FP	Different Names	Different Unique Code	Same Unique Code	Wrong
TN	Different Names	Different Unique Code	Different Unique Code	Correct

- Python 3.6.6
- Anaconda 5.2
- PyCharm Community Edition 2018.2.2

4.5.1.1 Working Environment

We used a Linux machine for the experiments, which was running in 64-bit Ubuntu 18.04 LTS operating system. It has an Intel Core i5-6200U 2.80 GHz CPU, 8.00 GB RAM and 1TB hard disk.

4.5.1.2 Algorithm Implementation Information

We have implemented our nameGist algorithm in Python programming language. We have collected the Python implementation of Modified Name Significance algorithm from its authors [100]. We use the open source Jellyfish [157] Python package for Soundex, Metaphone, NYSIIS and Match Rating Codes algorithm.

4.5.1.3 Measurements

At first, we are defining the matrices to compare the performance of the phonetic algorithms. The confusion matrix is described in Table 4.12.

True Positive (TP): For two same name, if an algorithm gives the same unique code, then it is a True Positive.

False Negative (FN): For two same name, if an algorithm gives different unique codes, then it is a False Negative.

False Positive (FP): For two different names, if an algorithm gives the same unique code, then it is a False Positive.

True Negative (TN): For two different names, if an algorithm gives different unique codes, then it is a True Negative.

We used different types of datasets and measured accuracy rate, precision, recall and F1 score of Modified Name Significance (M. Name Sig), Soundex, Metaphone, NYSIIS, Match Rating Codes, and nameGist algorithms. [200].

Accuracy Rate: Accuracy is the most commonly used performance measure. It is the ratio of correct output from the total output.

$$AccuracyRate = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

Precision: Precision determines how precise or accurate a model is. It finds out, from the total number of positive results that an algorithm returns, how many are actually positive. The high precision score means low ambiguous result.

$$Precision = \frac{TP}{TP + FP} \quad (4.2)$$

Recall: Recall, also known as Sensitivity or True Positive shows how many actual positive results are labeled as positive by an algorithm.

$$Recall = \frac{TP}{TP + FN} \quad (4.3)$$

F1 Score: The F1 score is known as the harmonic mean of precision and recall. It is the weighted average of Precision and Recall.

$$F1Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4.4)$$

Accuracy sometimes can be misleading. Because it only counts true or correct results. F1 Score is more useful than accuracy when the false and ambiguous measurement has importance. In record linkage and other application of the phonetic algorithms, it is essential that an algorithm should give correct result as well as have higher precision and recall value. So F1 score is a better measure to determine the performance of a phonetic algorithm.

4.5.2 Results

Below we discuss the results obtained by our compared algorithms: Modified Name Significance, Soundex, Metaphone, NYSIIS Match Rating Codes, and nameGist for all different types of datasets.

4.5.2.1 Bengali Name in English Representation (Bengali Phonetic)

The results for the Bengali phonetic dataset of Section 4.4.1 is presented in Table 13. From the Table 4.13, we see our algorithm has the best F1 score and accuracy score. Though Metaphone has 100% precision but has only 65% recall score. We also see that Soundex has a very low precision score as it only considers the first part of a name. Modified Name Significance achieved second highest F1 score (86.92%) with worst time requirement among all the algorithms.

4.5.2.2 English (British/American) Names

We have used two datasets of US/UK names (details in Section 4.4.2). Comparison of the algorithms in terms of accuracy and F1 score for these datasets are shown in Figure 4.4 and Figure 4.5. It is clear that nameGist achieved highest accuracy and F1 score in both cases compared with popular English phonetic algorithms and existing Bengali phonetic algorithms. Detail results are presented in Table 4.14 and Table 4.15.

Table 4.13: Result of Bengali phonetic name dataset

	Precision	Recall	F1 score	Accuracy	Time (sec)
M. Name Sig	99.12%	77.40%	86.92%	99.06%	1.48
Soundex	63.98%	92.47%	75.63%	97.60%	0.19
Metaphone	100.00%	65.41%	79.09%	98.61%	0.18
NYSIIS	70.24%	80.82%	75.16%	97.85%	0.19
Match Rating Codex	99.07%	72.95%	84.02%	98.88%	0.18
nameGist	99.22%	86.99%	92.70%	99.45%	0.44

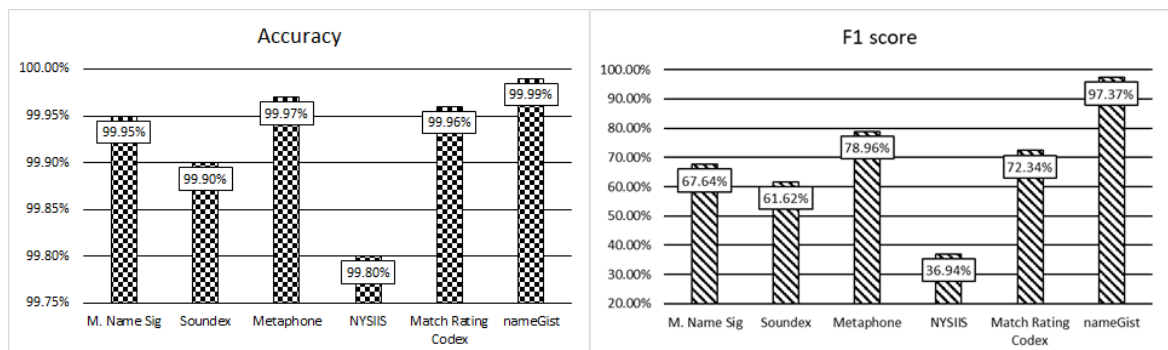


Figure 4.4: Accuracy and F1 score of all algorithms for English name dataset 1

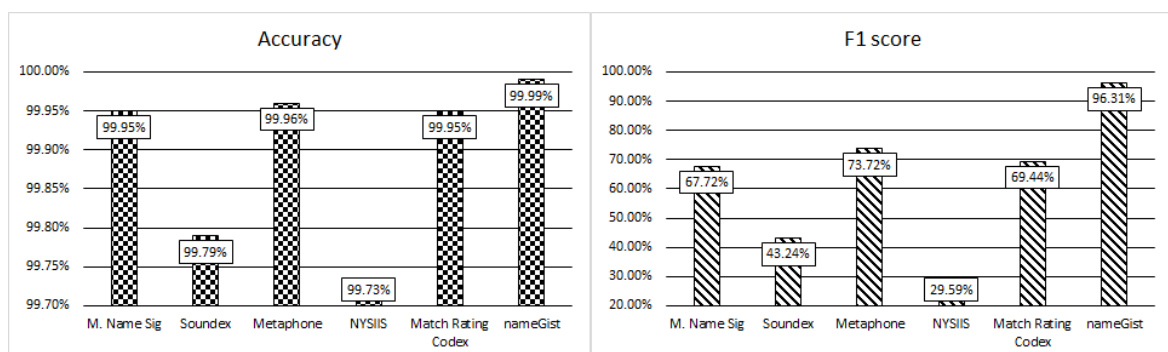


Figure 4.5: Accuracy and F1 score of all algorithms for English name dataset 2

Table 4.14: Result of English name dataset 1

	Precision	Recall	F1 score	Accuracy	Time (sec)
M. Name Sig	100.00%	51.10%	67.64%	99.95%	769.46
Soundex	48.59%	84.17%	61.62%	99.90%	49.45
Metaphone	99.85%	65.30%	78.96%	99.97%	50.54
NYSIIS	26.69%	59.98%	36.94%	99.80%	47.47
Match Rating Codex	99.65%	56.77%	72.34%	99.96%	46.55
nameGist	99.69%	95.15%	97.37%	99.99%	192.56

Table 4.15: Result of English name dataset 2

	Precision	Recall	F1 score	Accuracy	Time (sec)
M. Name Sig	100.00%	51.20%	67.72%	99.95%	767.61
Soundex	29.77%	78.97%	43.24%	99.79%	49.24
Metaphone	100.00%	58.38%	73.72%	99.96%	50.35
NYSIIS	20.00%	56.90%	29.59%	99.73%	46.8
Match Rating Codex	99.16%	53.42%	69.44%	99.95%	46.24
nameGist	99.89%	92.97%	96.31%	99.99%	195.94

Table 4.16: Result of Bengali unicode dataset

	Precision	Recall	F1 score	Accuracy	Time (sec)
nameGist	98.00%	99.53%	98.76%	99.99%	34.14
M. Name Sig					
Soundex					
Metaphone					Do Not Support Bengali Unicode
NYSIIS					
Match Rating Codex					

From the two results of Table 4.14 and Table 4.15 we see, our algorithm nameGist gives the best F1 score, accuracy score and Recall score. Modified Name Significance and Metaphone gives around 100% precision, but both of them have low recall score. NYSIIS and Soundex scored very low precision scores, because of their high ambiguity results.

4.5.2.3 Bengali Name in Bengali Representation (Bengali Unicode)

This dataset contains only Bengali Unicode names. For example তানভীর রহমান, তানবির রহমান, তানবীর রহমান etc. (details in section 4.3). Results are presented in Table 4.16. As other algorithms do not support Bengali Unicode, so we could not provide any result for them. For Bengali Unicode dataset, our algorithm gives 98% F1 score and 99.99% accuracy score.

4.5.2.4 English and Bengali Mix Dataset

This dataset consists of Bengali Unicode, Bengali Phonetic and English names (details in section 4.4.4). Results are presented in Table 4.17. As other algorithms do not support Bengali Unicode, they could not process the mixed dataset. We can see that our algorithm gives excellent accuracy (99.99%) and F1 score (97.03%) with the combined dataset.

4.5.3 Discussion

We have measured the performance of our algorithm nameGist in four scenarios, i.e., a dataset containing only Bengali phonetic, a dataset containing only Bengali Unicode, a

Table 4.17: Result of English and Bengali mix dataset

Algorithm Name	Precision	Recall	F1 score	Accuracy	Time (sec)
nameGist	99.66%	94.54%	97.03%	99.99%	268.60
M. Name Sig					
Soundex					
Metaphone					Do Not Support Bengali Unicode
NYSIIS					
Match Rating Codex					

dataset containing only English phonetic (British/American), and a dataset containing a mixture of these three types. We found that nameGist could handle all the four datasets properly and achieved higher scores in terms of accuracy and F1 score. In all the mentioned cases, our algorithm gives the best F1 score compared to other popular English phonetic algorithms and existing Bengali phonetic algorithms.

Other algorithms could not process Bengali Unicode dataset of section 4.4.3 and mixed dataset of section 4.4.4, because none of them supports Bengali Unicode. So it is a significant advantage of nameGist over other existing phonetic algorithms to be able to process Bengali Unicode.

We have compared nameGist with popular phonetic algorithms of two types.

- Type 1: Those algorithms which have been developed for English names, e.g., Soundex, Metaphone, NYSIIS, and Match Rating Codes.
- Type 2: The algorithm intended for Bengali names in English, e.g., Modified Name Significance.

Algorithms of type 1 require less time to process a dataset than nameGist for phonetic encoding. We are explaining the reason here. The nameGist supports Bengali Unicode names which the other algorithms do not support. To support Bengali Unicode, in the nameGist algorithm, we have included two additional steps.

1. Check whether an input name is in English or Bengali Unicode (at the beginning of the algorithm)

2. Convert to equivalent English phonetic code, if the input name is in Bengali Unicode

Above points are discussed elaborately in section 4.3.3. For the above two additional steps, nameGist requires more time to process a dataset than the Type 1 algorithms. In return, nameGist gives much better accuracy and F1 score than Type 1 algorithms for Bengali phonetic name dataset. Besides, only nameGist can process Bengali Unicode dataset, which could not be done by its competitors.

4.5.4 Limitation

A question may be arisen that the step of changing all vowels to a to minimize noise and to improve performance of matching may disturb the phonetic composition. The answer of the question is that vowels are important for phonetic compositions, and removing vowels may disturb phonetic composition. However, the fact is that this is how phonetic algorithms work for record linkage, entity resolution, name matching, etc. The most popular phonetic algorithms nowadays, e.g., Soundex, NYSIIS, Metaphone, and Double Metaphone, also works in the same way. Because of the ambiguity in representing vowels and the presence of a high amount of noise in any real dataset, it is impossible to perform record linkage or name matching properly without removing vowels or encoding them. The primary application areas of phonetic algorithms are the census, financial, and health sectors, where correct name matching is required with the presence of noise. The ambiguity of presenting vowels in written form compels the phonetic algorithms to remove or encode vowels during generating phonetic codes from names. Another significant thing to mention is that phonetic algorithms are mostly used in the intermediate step of record linkage or name matching to improve performance. Thus the primary (or base) datasets, among which record linkage is performed, remain unchanged in their first place. Thus, in most cases, phonetic compositions of the primary datasets will not be disturbed.

Another question may be arisen that, practically some records may not have all sub-names or sub-names of a name could be in shuffled locations. Then how nameGist will respond to those names. The answer of the question is that, the nameGist is a phonetic algorithm, and by definition, it matches two names or words by their similarity of pronunciation. If we remove sub-names or shuffle sub-names, than the names will no longer be similar to

pronounce. That is why such records were not presented as examples in the paper. However, it should be noted that, if a name has only one part, e.g., Sabuj, nameGist will generate correct phonetic code for it.

Another question may be arisen that, what about different people with similar sounding names with different spelling. For example: Ibrahim vs Abraham. We inputted these examples in nameGist and Soundex and did a comparison. From the comparison, it is precise that both nameGist and Soundex process most of these name pair as same. Because these name pairs are very similar to pronounce. In entity resolution, record linkage, and similar application areas, where the phonetic algorithm fails to provide correct results, other parameters such as age, gender, address, etc. of the persons play an essential role in differentiating between the persons. The same thing is applicable vise versa. If same person's name is being inputted in a database with different spelling, nameGist may generate different codes for them, if their pronunciation is also different. This is absolutely logical. The record linkage system still might be able to link them as same persons records as the system will consider various other attributes for linkage such as age, zip code, gender, etc.

Table 4.18: Similar sounding names with different spellings

Input	Algorithm	Result	Verdict
Samrat, Simrit	nameGist	smrt, smrt	Similar
	Soundex	S563,S563	Similar
Ibrahim, Abraham	nameGist	abrhm,abrhm	Similar
	Soundex	I165,A165	Different
Grind, Grand	nameGist	srnd,srnd	Similar
	Soundex	G653,G653	Similar
Gate, Gait	nameGist	st, sat	Different
	Soundex	G300,G300	Similar

The nameGist algorithm detects and removes vowel marks. Then it maps the similar-sounding words to one. However, this approach does not give the correct result in some

cases. For example, Momo (মম) vis-a-vis Mimi (মিমি): These two are different names with some similarity to pronounce. Both have the same core letters with a change in the vowel marks only. nameGist will consider these two names as one and will generate the same output, which is incorrect. However, this failure is not only associated with nameGist. As other phonetic algorithms, e.g., Soundex, Metaphone, also remove vowel marks to reduce ambiguity, they will also encounter such kind of failure. Table 4.18 presents some other examples for which nameGist and Soundex fails. The error rate of nameGist is significantly lower than other popular phonetic algorithms, which can be easily understood from Table 4.13 - 4.15. For example, from Table 4.13, we can see that the error rate of nameGist is only 0.55%, whereas, for Modified Name Significance, Soundex, and Metaphone, the error rate is 1%, 2.4%, and 1.4% respectively.

Table 4.19: Examples of failures

Input Name Pair	Algorithm	Result	Verdict
NASHIT, NISHAT	nameGist	nst,nst	Incorrect
	Soundex	N230,N230	Incorrect
RUMA,RUMI	nameGist	rm,rm	Incorrect
	Soundex	R500,R500	Incorrect
MAZEDA, MASUDAă	nameGist	msd, msd	Incorrect
	Soundex	M230,M230	Incorrect
ALIM, ALAM	nameGist	alm,alm	Incorrect
	Soundex	A450,A450	Incorrect

Considering all the results, we can say that, the nameGist algorithm is highly practical as it can handles Bengali Unicode, Bengali Phonetic, English Names at the same time. Though nameGist takes higher time, but it will be well accepted by the users as in most application areas of phonetic algorithm e.g., name matching, record linkage, entity resolution, where greater accuracy is highly desirable.

4.6 Conclusion

In this chapter, we have proposed a new algorithm “nameGist” for phonetic encoding. The nameGist is a novel algorithm because it is the first algorithm that supports Bengali phonetic in English language, Bengali language, English phonetic and a combination of all. Our algorithm can handle noisy data and can encode misspelled names properly. We have compared the performance of nameGist with popular phonetic algorithms for English and Bengali, e.g., Soundex, Metaphone, NYSIIS, Match Rating Codes, and Modified Name Significance using four datasets. Experimental results show that, for all the datasets, nameGist performs better in terms of accuracy and F1 score. Bengali is gaining importance day by day for storing records in the computer. More than 250 million people, around the world, speak in the Bangali, which is the seventh most spoken language in the world. As nameGist can process Bengali Phonetic in English and Bengali Unicode along with English phonetic, it will be very helpful for record linkage, name-matching, database indexing, etc. in the cases where the Bengali language is used. In the future, the work can be extended to support other languages.

Chapter 5

Key-based Secured Record Linkage

In this chapter and also in the next chapter, we investigate our third research problem that is record linkage. Nowadays, a large amount of health data is electronically accessible, available, and processable. In developed countries, health data can be integrated using a social security number or national health id. However, in developing countries such as Bangladesh, health data integration is a very challenging task due to noisy, incomplete, and missing data. Another challenge is the ambiguity in patient identification due to the absence of standard patient identification key. In this chapter, we introduce a technique namely Key-based Secured Record Linkage (KSRL). Experimental results on real health data show that our KSRL technique can link records smoothly with high precision, recall, and F-measure in the presence of noise and the lack of universal health ID.

We structure the rest of this chapter as follows. In Section 5.2, we discuss the importance of health data integration. The privacy and safety issues related to healthcare data are briefly explained in Section 5.3. Then in Section 5.4, we review socioeconomic characteristics and their effect on medical data for developing countries. Section 5.5 discusses the healthcare data formation process in Bangladesh, as a case study of developing countries. In Section 5.6, we propose the Key-based Secured Record Linkage (KSRL) technique. The performance analysis of KSRL is presented in Section 5.7. Lastly, Section 5.8 concludes this chapter.

5.1 Introduction

Health data integration is essential for better health outcomes. Every day thousands of patients visit public and private hospitals, pathology centers, doctors' chambers. A lot of health records and related files are generated there. These valuable medical data are stored in different health management systems such as Picture Archiving and Communications schemes, Clinic Information schemes, Healthcare Information System, etc. Because of the scattered information in different systems, useful knowledge discovery cannot be achieved. The proper integration of scattered healthcare data in a warehouse is needed [2,6,107] .

To get the best benefit from diversified healthcare datasets, the linkage of records is necessary. Record Linkage is the way of finding record pairs, belong to the same entity or person in the real world, from various datasets. Given two or more datasets, the record linkage process discovers the twins that are alike [41,48]. Application areas of record linkage include healthcare, finance, census, etc.

Healthcare data that contains protected health information (PHI), e.g., the patient's name, address, birth date, etc. can be linked easily. However, healthcare data containing PHI is lucrative to hackers, and the sale value of these data is very high [109,166]. So, to protect these data from hackers or other malicious parties, Privacy Preserved Record Linkage (PPRL), is one of the significant research spotlights [183].

Some researches have been performed regarding record linkage based on social security number (SSN) or some identification keys [66,110,124]. However, in many countries such as Bangladesh, hospitals do not use National ID numbers or other identification keys to store patient data. So those approaches are not effectively applicable in Bangladesh or other economically developing or under-developed countries for record linkage. Illiteracy is another issue of getting quality healthcare data. Many peoples do not share their full name and birth date correctly. Thus, data stored at healthcare centers of developing countries are noisy and difficult to integrate [104].

To address the record linkage problem in the absence of a National ID, we first propose Patient Identification Technique based on Secured Record Linkage (PITSRL) [107]. The input of PITSRL system is health records provided by different health care

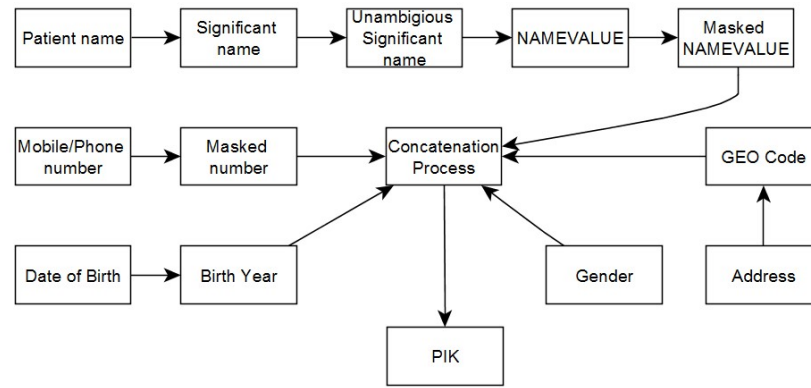


Figure 5.1: Block diagram of PITSRL

organizations such as government and private hospitals, diagnostic centers, research centers, health NGOs. These raw health records contain attributes related to patient identification such as patient name, address, and mobile number. PITSRL algorithm works in two steps.

- In step 1, a Patient Identification Key (PIK) is generated for each record using available patient identifiable data.
- In step 2, all identifiable data, capable of identifying individual patients, are removed from each health record.

We used five attributes to generate identification key in PITSRL. Those are: mobile number, name, age, geocode, and gender. Mobile numbers are made secured through masking. Name is converted to NameValue. Age is used to generate the year of birth and age group. Block diagram of the system is presented in Figure 5.1. Details can be found in [107]. The technique is useful for record linkage in the absence of National ID or SSN. A limitation of PITSRL is, it used mobile number of patients for generating the key. However, mobile number of same patient varies with time and some patients e.g., infants do not have mobile number.

In this chapter, we propose a PPRL technique, Key-based Secured Record Linkage (KSRL) that link records efficiently without National ID and in the presence of noisy data. Empirical results show that KSRL can effectively connect records in the scarcity of universal ID numbers and the availability of erroneous data, e.g., misspelled of patient Name. This research is an extension of our ongoing research on Patient Identification

Technique based on Secured Record Linkage (PITSRL) [107]. KSRL has the following new contributions which were not present in PITSRL:

1. We have categorized the patient identifiable attributes into three categories: changeable attributes, fixed unambiguous attributes, and fixed ambiguous attributes. Then we exclude changeable attributes from generating record linkage keys. By this, we could solve different issues of our previous technique “PITSRL,” such as changes in mobile numbers of patients.
2. We have added MD5 based hashing for name and address attributes for generating keys. Thus KSRL provides one extra layer of security than PITSRL.
3. We have introduced techniques for evaluations of the privacy of KSRL to justify whether KSRL is practically implementable or not. Results presented in a new section called “Privacy Evaluation.”
4. The accuracy of proposed KSRL techniques is almost 97%, which is much higher than the previous technique PITSRL, which was 87%.

5.2 Proposed Solution: Key-based Secured Record Linkage (KSRL) Technique

In this section, we describe our proposed technique, KSRL comprehensively. From different health service providers, we have collected healthcare records contained identifiable patient attributes, e.g., name, date of birth address, gender, mobile number, height, weight, test result, etc. We categorize identifiable patient attributes into three categories: changeable attributes, fixed unambiguous attributes, and ambiguous attributes.

5.2.1 Changeable Attributes

From the perspective of Bangladesh, patients who visited the hospitals, clinics, diagnostic centers, or other health care centers, many of them are not literate. We

found that there are some changeable attributes in health records provided by different healthcare centers. The present address of the patient can be changed because he can leave the current place where he lives. Some people do not know their age correctly because they did not register their birth certificates. For several years, a patient provides unchanged age (e.g.,50 years) to healthcare centers. A large number of people use multiple mobile numbers. Such patients may provide a mobile number in one healthcare center and a different mobile number in another healthcare center. Similarly, they can provide a different mobile number in the same center at different times. Weight and height of patients are changeable with time. So, we listed the changeable attributes they are: present address, age, mobile number, height, weight, test result, etc. We did not use these changeable attributes for developing record linkage key in our Key-based Secured Record Linkage (KSRL) Technique.

5.2.2 Fixed Unambiguous Attributes

Fixed unambiguous attributes are those which are not changeable as well as give precise information about a patient. We mentioned the Patient's Gender and Date of Birth (DOB) to fixed unambiguous attributes. Gender cannot be changed, and the date of birth will remain the same. We convert Gender to Gender Code as seen as Table 5.1. We then convert the DOB to Birth Year Range.

Table 5.1: Gender code

Gender	Gender Code
Male	M
Female	F

In Table 5.2, we convert Date of Birth to Birth Year, then convert Birth Year to Birth Year Range for k-anonymity. We also convert DOB to Age Range. Age Range chart is given in Table 5.3.

In Table 5.4, we convert a date of birth to birth year and then convert the birth year to age, and lastly, we convert the age to age range by the age range chart of

Table 5.2: Conversion of date of birth to birth year range

Date of Birth	Birth Year	Birth Year Range
7/13/1992	1992	1990-1994
4/15/1978	1978	1975-1979

Table 5.3: Age range

Age Group Name	Age Range
Child	0-9
Teenager	10-19
Adult	20-59
Senior	60 and over

Table 5.4: Conversion of date of birth to age range

Date of birth	Birth Year	Age	Age Range
9/15/1994	1994	25	Adult
3/15/2009	2009	10	Child
1/13/2005	2005	14	Teenager
7/20/1952	1952	67	Senior

Table 5.3. The age range is needed for future data analysis tasks such as which age range patients have back pain, fatigue, AIDS, measles, etc. After that, we concatenate the Gender Code and Birth Year Range attributes. If someone's Gender Code is M and Birth Year Range is 1990-1994, the concatenation result is M1990-1994. Then we apply the hashing (MD5) algorithm. The output of the hashing algorithm will be used in the next step by the KSRL algorithm. Table 5.5 shows the processing of the fixed unambiguous attributes.

Table 5.5: Fixed unambiguous attributes

Gender Code	Birth Year Range	Concatenated Value	Hashed (MD5) Value
M	1990-1994	M1990-1994	2c7e8872665fe05aef7 c8c2ba4099d63
F	2005-2009	F2005-2009	ca0d657199fb6fd4d3d2a 957ceee207c

5.2.3 Ambiguous Attributes

Patients provide their names and addresses in a different format at different health care centers. In some places, they give their nickname, and in some other place, they give the full name. Sometimes data entry operators input patient names differently than their real names. Patients' addresses also faced ambiguity because of the same reason as patients' names. . That is why we separate these two attributes as ambiguous attributes. To remove the ambiguity that is to reduce the noise in the patient name and address values, we have used popular phonetic algorithms such as Soundex and Metaphone. We also used our developed algorithm NameValue which is presented in the Section 4.2.2. We compare the performance of NameValue, Soundex, and Metaphone algorithms. We run 1000 patient names in java code of NameValue, Soundex, and Metaphone. A name table of 1000 patient name is collected from a local hospital. In 1000 patient name, 908 patient names are distinct, and 92 patient names are repeated. We find that total run time of NameValue algorithm is 0.9076 seconds (10 runs) and used memory is 8397272

bytes, total run time of Soundex algorithm is 0.774 seconds (10 runs) and used memory is 7858832 bytes, total run time of Metaphone algorithm is 1.1965 (10 runs) and used memory is 7886896 bytes.

We create a dataset of 1000 patient names differently and calculate accuracy. We find NameValue algorithm accuracy is 90%, Soundex algorithm accuracy is 58.9%, Metaphone algorithm accuracy is 77.8%. For this above reason, we find that NameValue is the best for masking Patient Name.

We create a dataset of 1000 Address and calculate accuracy. We find NameValue algorithm accuracy is 93.667% and total run time is 1.1616 seconds (10 runs), Soundex algorithm accuracy is 95.833% and total run time is 0.9332 seconds and (10 runs), Metaphone algorithm accuracy is 95.1667% and total run time is 1.2214 seconds (10 runs). For this above reason, we choose the Soundex algorithm for masking the Patient's address.

To make the patient name and address secure, we convert the patient name to NameValue and then apply the hashing algorithm (MD5). We have applied the Soundex and MD5 algorithm, respectively, to the patient address.

5.2.4 NameValue Generation

We remove the less significant or insignificant portion of the Name, i.e., salutations and titles (we use 201 salutations) of the name and then separate the significant portion. Then the output string is masked through our Code Table. Then we apply the hashing algorithm MD5 to this masked NameValue. The NameValue formation process is illustrated in Table 5.6, and the algorithm is presented in Section 4.2.2.

Soundex is applied to the patient's addresses, and then hashing (MD5) is applied to disguise real addresses from the users. The process of the hashed address is explained in Table 5.7.

5.2.5 KSRL Key Generation

Our proposed algorithm namely Key-based Secured Record Linkage (KSRL) is shown in Algorithm 5.1. The block diagram of the KSRL key generation system is shown in Fig.

Table 5.6: Illustration of significant ambiguous hashed value selection

Patient Name	Significant Portion	Unambiguous Significant Portion	Masked NAME VALUE	HASHED (MD5) VALUE
Mr. Abu Naser	Abu Naser	ab nsr	tasjml	96f5e9f915715a99f cfa3008248fa21
Mr. Md. Abu Naser	Abu Naser	ab nsr	tasjml	96f5e9f915715a99f cfa3008248fa21
Mohammad Abu Naser	Abu Naser	ab nsr	tasjml	96f5e9f915715a99f cfa3008248fa21
Mr. Soumik Chakraborti	Soumik Chakraborti	smk chkrbt	migsbeglaln	6e811887e983b437f 89cb42c9503cc74

Table 5.7: Hashed address

INPUTTED ADDRESS	SOUNDEX	HASHED (MD5) VALUE
Muradpur, Chittagong	M631	2af15281c6482f61aee0d88dba3d362b
Satkania, Chittagong	S325	9ff7ebe4b256a14abd093d464be6a9ea
Bhola, Barisal	B416	cf843909e20d608f91ef3428c0746616
Muktagasa, Mymensingh	M232	e35b0f257c3d4fccdf275f679c3d370a

5.2.

Algorithm 5.1: KSRL**Input:** Health record with PHI**Output:** Anonymized health record with KSRL-Key

```

1 while !End of Recordset do
2   Convert patient Name to NameValue, then apply hashing (MD5) algorithm;
3   Convert Gender to Gender Code;
4   Convert Birth Date to Birth Year Range;
5   Concatenate Gender Code and Birth Year Range, then apply hashing(MD5)
   algorithm;
6   Convert Address to Soundex, then apply hashing (MD5) algorithm;
7   Generate Key-based Secured Record Linkage (KSRL) from Concatenation of
   above attribute;
8   Add KSRL to the recordset;
9   Remove Patient Name, Date of birth, Address, Gender data;
10 end

```

Table 5.8: Sample KSRL-key

KSRL-Key
EDC6333B5A73DC7219145406E4BEE8F36F8F8C454E5D80D64 A76643E2833E9B22AF15281C6482F61AEE0D88DBA3D362B
A69CC2DD8E550BB3F2609F32C6227C52572AB12BF966406AA 0FF0829A87229A19FF7EBE4B256A14ABD093D464BE6A9EA
65A8EA2179F62342D2D9CA645B72E90F2C7E8872665FE05AEF 7C8C2BA4099D6319056A4158E72E7EAD6F5F1FF03FB95B
0FD3C989098473FCC40465555E2DC136C65CB6F948F979B360 DEE8AD247B93C56F72A4E2135163AA882D662EF7245E3A

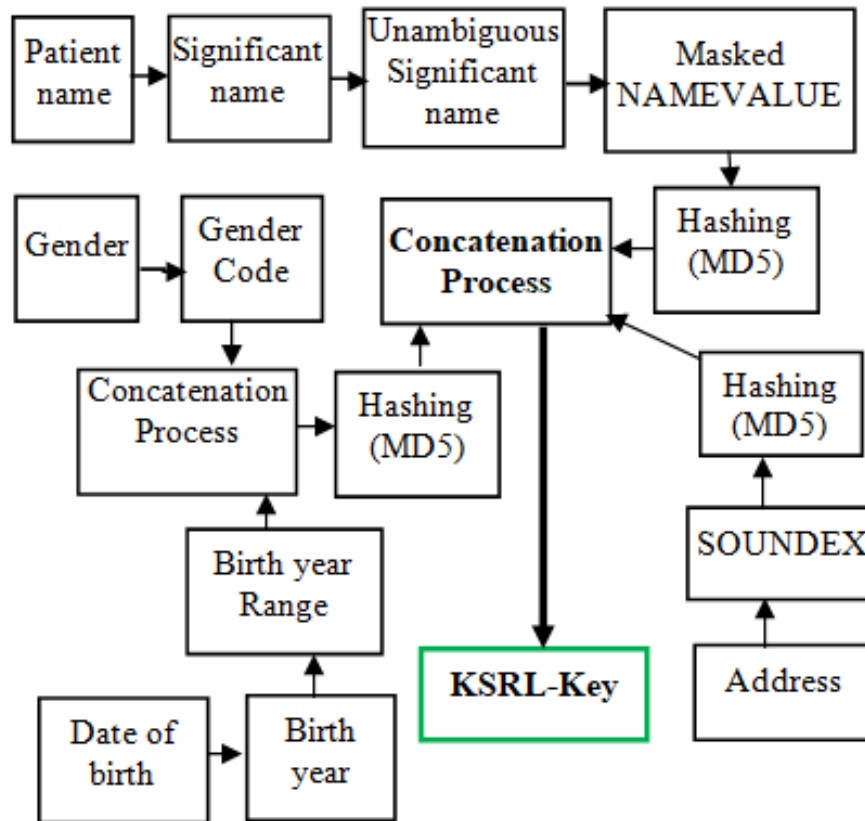


Figure 5.2: Block diagram of key-based secured record linkage

In Table 5.8, KSRL-Key, a 96-bit key, is the output of our Key-based Secured Record Linkage (KSRL) algorithm. We assume the name attribute as 40-bit, gender as 1-bit, birth year range as 9-bit, and address as 40-bit. So, our total original input size is 90-bit. Our KSRL output size, after concatenation, is 96-bit.

5.3 Results and Discussion

We compared the performance of NameValue, Soundex, and Metaphone algorithms, and we found that NameValue is the best for masking the patients' names. The total runtime of the NameValue algorithm is 0.9076 seconds for ten runs, used memory is 8MB, and the accuracy is 90%. For encoding patients' addresses, Soundex performed better than NameValue and Metaphone. For address, Soundex get 95.833% accuracy with a total runtime of 0.9337 seconds (10 runs).

We collected 10,000 patient records from a healthcare center with proper ethical permission. Patient name and Gender are extracted from the dataset. Date of Birth are

created using `RANDBETWEEN (DATE (1960,1,1), DATE (2017,1,1))`. Addresses have been preprocessed by following [138]. Our dataset contains four attributes, e.g., Patient Name, Address, Gender, Date of Birth. We used Java to implement our KSRL algorithm. The total run-time is 1.9126 seconds (10 runs). Used memory is 10200368 bytes. The size of the input file was 2187 KB. The size of the output file was 1087 KB. We define the performance matrices in our context as follows. Accuracy of a system is the division of correct matches by total number records. Precision is the part of record-set, classified as matches, by a model that are real matches. Recall (also called sensitivity) is the part of real matches that are precisely classified as matches by a model. We found accuracy 96.42%, precision 93.3%, recall 96.4%, and F-measure 94.8%.

We built another dataset by randomly selecting 500 records from the patient dataset to manually check whether KSRL can identify the same patients correctly or generate different record linkage keys for a single patient. The later dataset contains health records of 166 distinct patients. Table 5.9 represents the details of our selected dataset with data repetition information (how many records of the same patient in the dataset of 500).

Table 5.9: Patient dataset analysis

Description of Reception	No. of Patients	No. of Records
Ten records of the same patient	17	170
Eight records of the same patient	15	120
Four records of the same patient	15	60
Three records of the same patient	11	33
Two records of the same patient	9	18
Single record of a patient	99	99
Total	166	500

After inputting the above dataset to KSRL, it generated 166 distinct KSRL-key. Moreover, our system generated the same key for the same patient with marginally

misspelled names and addresses. For very few cases, it generated different keys for the same patient and the same key for different patients. This error is due to the high similarity in the patients' data and noise.

5.3.1 Clustering Analysis

Clustering is a method of partitioning a collection of records (or objects) into a collection of significant sub-classes, called clusters. Hierarchical clustering is a process of cluster analysis which tries to make a hierarchy of clusters. We used agglomerative hierarchical clustering or bottom-up approach for cluster analysis. We used the Euclidean distance function for distance measure. We inputted 500 records of 166 patients as input, and we found 166 output of cluster instances. We found accuracy 97%, precision 94.6%, recall 97%, and F-measure 95.7%.

5.3.2 Finding the Significance of the Attributes

To analyze which attribute has higher impact on the accuracy of KSRL, we removed one attribute at a time, then generate record linkage key without that attribute. After that, we perform the record linkage and calculate accuracy. We repeated this step for the following attributes.

- Name
- Gender
- Address
- Date of birth

The input dataset sample is presented as Table 5.10. The impact of each attribute over accuracy is shown in Table 5.11. We can see that, if we remove the "Birth Year Range" attribute and generate key using other attributes, the accuracy decreases 6.9% which is highest. So, we can say that, "Birth Year Range" attribute has the most significance impact on record linkage accuracy.

Table 5.10: Sample records for key generation

Id	Name	Gender	Birth Year Range	Age
0	MRS. TASHNUVA	Female	2000-2004	18
1	MR. A. JALIL	Male	1985-1989	33
2	MR. AL AMIN	Male	1985-1989	33

Table 5.11: Impact of attributes for record linkage

Excluded Attribute from Key Generation	Accuracy Decrease
Name	0.20%
Address	0.60%
Gender	6.60%
Birth Year Range	6.90%

5.3.3 Effects of Privacy Preservation over Record Linkage Performance

To analyze the effects of different privacy preservation techniques that we have used in KSRL, we performed the following procedure. We concatenated the raw attribute values from the dataset to use it as the linkage key. This time we did not use any privacy preservation method such as phonetic encoding, generalization and hashing. We found that, without privacy preservation, if we use concatenated string as the KSRL key, the accuracy of linkage increase almost 3%. That is, for the raw key, we found accuracy 99.5%. Record linkage time increase to 4.7 times higher than previously processed privacy preserved dataset. We can easily justify both. As, in the case of privacy preservation, actual data is modified, which produce an adverse effect on accuracy. Again, as generalization and phonetic encoding simplify the raw data, KSRL runs much faster, using privacy preserved dataset. These are really interesting findings for PPRL.

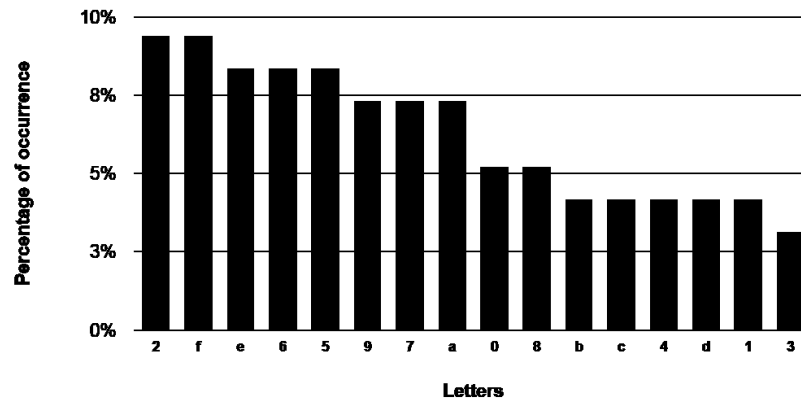


Figure 5.3: Frequency analysis

5.3.4 Privacy Evaluation

In privacy analysis, we have used a popular privacy evaluation techniques known as frequency analysis. We have applied frequency analysis to KSRL to find out whether the re-identification of the patients from KSRL-Key is possible or not.

Frequency analysis: Frequency analysis is the study of the frequency of characters or combinations of characters in a cipher-text. The process is utilized as an aid in defeating classical ciphers. If the frequency distribution of hash-encoded values, proximately matches the distribution of values in a (public) database, then ‘re-identification’ of values might be possible. After frequency analysis, the re-identification of original values might not be possible because we use the MD5 hashing algorithm and some phonetic algorithms. We describe frequency distribution of the keys generated by KSRL in Fig. 5.3.

From Figure 5.3, we see the frequency distribution of our generated KSRL keys. Letters and numbers of KSRL are not original letters and numbers. These letters and numbers are encrypted letters and numbers. So, we can say that using the frequency analysis, re-identification of original values might not be possible.

Dictionary attack: A dictionary attack is usually carried out to break passwords or similar encrypted data in a database with the help of an available digital dictionary. Dictionaries contain so many words that usually many people's password matches from there. An attacker uses some tools to automatically go through all the words on a

dictionary and find a word to decrypt the data. The keys, generated by KSRL for record linkage, are safe from dictionary attacks. The reason is evident as we have used multiple privacy techniques such as phonetic encoding, generalization, one-way hashing, so our generated key can not be broken by a dictionary attack.

5.4 Limitation

From the Result and Discussion Section we can see that, our proposed approach KSRL can perform record linkage efficiently in the absence of a global patient identifier such as National ID, Health ID or Social Security Number. A limitation of our proposed method is that to preserve the privacy of the sensitive health data, we have performed MD5 hashing on the concatenated record linkage key. So KSRL falls in the category of exact matching, which is less flexible than approximate matching. Exact matching is highly sensitive to errors. For example, if a patient's name is inputted with a little typographical error, his or her hash value will be changed entirely. To handle noise in data, in KSRL, before hashing, we have performed phonetic encoding on the Name and Address attribute and also performed generalization of Date of Birth attribute. These steps add flexibility and robustness of our proposed technique to deal with noisy data. In the next chapter, we propose another privacy-preserving record linkage technique that adopts approximate matching.

5.5 Summary

In this chapter, we provided a practical solution: Key-based Secured Record Linkage (KSRL) technique that can perform record linkage properly while maintaining the privacy of the protected health information. KSRL will make it easier for health service providers and analysts to discover hidden knowledge from healthcare data. We have used MD5 hashing to patients' names and addresses so that any malicious entity, hackers, and even users of a health data warehouse cannot find the real name and address of a patient. It is observed from the empirical results that KSRL could properly link records even when no standard ID number is available and also in the context

of noisy data. For a noisy real health dataset of ten thousand patients, with missing values, KSRL achieved 96.4% accuracy and 94.8% F-score. We have also evaluated the privacy preservation of our technique using frequency analysis and dictionary attack and found it safe. So, we can say that It is possible to share and incorporate information about patients', among different healthcare providers, using the KSRL technique.

Chapter 6

Incremental Record Linkage with Privacy Preservation

Using incremental approach to solve record linkage problem is a relatively new research area. In incremental record linkage, every inserted record is compared with some existing clusters of records based on its blocking key value. Then, considering similarity, either the record will be put into an existing cluster, or a new cluster will be created for it. Although few papers have presented their solutions for incremental record linkage targeting the linkage quality or efficiency, privacy issue regarding the approach has not yet been discussed. Privacy is a major concern when record linkage is performed for sensitive data, e.g., health records, financial records, etc. In this regard, we have come up with a novel concept privacy-preserving incremental record linkage (PPiRL) which encapsulates privacy-preserving techniques with incremental record linkage approach. In this chapter, we propose an end-to-end framework as our solution for PPiRL.

The rest of the chapter is organized as follows. In Section 6.2, we provide a review of the important literature related to record linkage, privacy preserving record linkage, and incremental record linkage. We provide some background and formulate the problem of privacy-preserving incremental record linkage in Section 6.3. Our main contribution, PPiRL framework, is explained in Section 6.4. Experimental results, privacy evaluation, and comparisons are presented in Section 6.5. Finally Section 6.6 concludes this chapter.

6.1 Introduction

Nowadays, fast-growing datasets that contain hundreds of millions of records are being collected, stored, processed, analyzed, and mined. To enable an in-depth analysis of such large datasets, information from multiple data sources is often required to be integrated. For getting maximum insight from integrated data (e.g., correlations among diseases in the case of a medical dataset), record linkage is necessary. Record linkage identifies the record pairs from various databases that belong to the same real-world entity, i.e., a customer, a patient. Given two datasets, the record linkage process finds out all record pairs that are similar to each other. Record linkage faces two challenges on the edge of big data. First, the high velocity of data updates swiftly makes previous linkage results extinct. Second, a massive volume of data requires a long time for applying record linkage in traditional (batch linkage) approach. These two challenges require an incremental solution so that when data updates appear, we can swiftly update linkage results [67].

Usually, distinct identifiers, e.g., primary keys, are not always present in the databases that need to be linked. This makes record linkage a problematic task. So, to perform linkage, the common attributes of datasets are used in many cases. These include name, birth date, address, and other personal details of an entity. Currently, maintaining privacy and confidentiality are significant challenges for record linkage. During the linking of databases across organizations using personal information, careful protection of the privacy of this information is a must. The process of discovering records of similar individuals from separate databases without disclosing identifying attributes of these individuals is known as ‘privacy-preserved linkage of records,’ ‘linkage of blind data,’ or the ‘private linkage of records’ problem [110,183,188].

Although few papers such as [38,55,67,126,175,192,193] have presented their solutions for incremental record linkage targeting the linkage quality or efficiency, privacy issue of this approach is yet to discuss. In this regard, we have come up with a new idea called “privacy-preserving incremental record linkage” or in short “PPiRL,” which encapsulates privacy-preserving techniques with an incremental approach to record linkage problems.

Our contributions are summarized as follows:

1. To the best of our knowledge, We are the first to recognize privacy-preserving incremental record linkage as a new field of research. Recognition of this field paves the way for solving the problems of record linkage, integration, data mining relating to volume and velocity of data along with privacy issues.
2. We have proposed a novel end-to-end framework that encompasses both privacy and linkage of data in an incremental approach. We have named it privacy-preserving incremental record linkage (PPiRL). We have also provided the necessary definition for PPiRL.
3. We have implemented our PPiRL framework and provide various comparisons of our framework with traditional privacy-preserving record linkage (PPRL) techniques and incremental record linkage (IRL) techniques.

6.2 Literature Review

6.2.1 Record Linkage

Record linkage [61] [77], schema matching [17] and data fusion [20] [17] are the three main tasks in data integration. Among them, record linkage is aimed at identifying all records that refer to the similar real-world entities in several databases. It can also be applied to detect identical records in a single database [58] [134]. Data quality plays a crucial role in record linkage. The fact that real-world data are ‘dirty’ is responsible for the loss of quality of linkage [76]. Only exact matching of personal identifying features is not enough for desired output. We need approximate matching besides accurate matching to achieve good accuracy [39] [50]. Usage of expensive similarity comparison methods creates a performance bottleneck [15] [43]. This challenge can be overcome by using proper indexing techniques [42]. Details of the steps used in record linkage are presented in Section 2.1 of Chapter 2.

6.2.2 Privacy Preserving Record Linkage

For knowledge discovery purpose, large databases across organizations needed to be linked. At the same time preserving the privacy of the records stored in these databases is also crucial. This necessity directs a new research area called privacy preserving record linkage (PPRL) [45] [188]. PPRL is alternatively called as privacy record linkage [4] [88] [198] [11] and blind-folded record linkage [45] [190]. Due to privacy concerns, commercial interests or legal restrictions, it is often not allowed to exchange private or confidential data between organizations. When there arises a cross-organizational project, databases of different organizations need to link in such a way that no sensitive information is being exposed to any of the parties involved, and no outsider can eavesdrop on the data to learn anything. PPRL ensures that after the end of a linkage project, only a limited amount of information is disclosed to the exchanging parties [183]. Details of the steps used in privacy preserving record linkage and different issues of PPRL are presented in Section 2.2 of Chapter 2.

6.2.3 Incremental Record Linkage

Incremental record linkage (IRL) is the clustering process where only the newly arrived records will be compared with existing clusters. Then, based on similarity, either the new records will be put into some existing cluster(s), or a new cluster will be created for it if the new records are dissimilar to existing clusters according to some threshold value. Incremental record linkage has been studied in [192] [193]; however, they focused on evolving matching rules and discussed concisely only evolving data.

On the other hand, incremental graph clustering methods have been proposed by some researchers. Mathieu et al. [126] studied incremental correlation clustering for the following two cases: (1) one vertex is added each time, and (2) already identified clusters need to be preserved. Charikar et al. [38] studied incremental clustering when the number of clusters is predefined. Both papers focused on theoretical analysis rather than implementations. A novel incremental heuristic algorithm was presented in [175] for the Clique Partition Problem (CPP), a well-studied graph partitioning problem. The algorithm was much faster for the tested datasets comparing batch linkage algorithm.

Privacy issues were not considered in any of the above researches.

An efficient approach for incremental record linkage has been proposed in [67] where the authors presented a framework using several algorithms and showed viable efficiency compared to the previous works. Nasciment et al. [55] proposed heuristic-based approaches to speed-up the performance of the IRL algorithm. Both the papers deal with linkage quality and efficiency. None of them considers the privacy issues for record linkage.

6.2.4 Summary of Literature Review and Research Gap

Record linkage is a widely studied research problem and different solutions have been proposed by the researchers since 1960s. Due to the increasing privacy issues, privacy preserving record linkage is also a very active research area since last two decades. Due to the increasing volume and update velocity of datasets in this big data era, IRL is resently proposed by few researchers to deal with scalability issues of traditional record linkage. But scalability support using incremental apporach along with privacy preservation of sensitive large-scale dataset are yet to deal with by the researchers. To the best of our knowledge, our framework is the first to perform an incremental linkage which considers the privacy issues. We will present our framework in Section 6.4. Before that, we formally formulate the Privacy Preserving Incremental Record Linkage (PPiRL) problem in the next section.

6.3 Background Knowledge and Problem Formulation

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

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

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

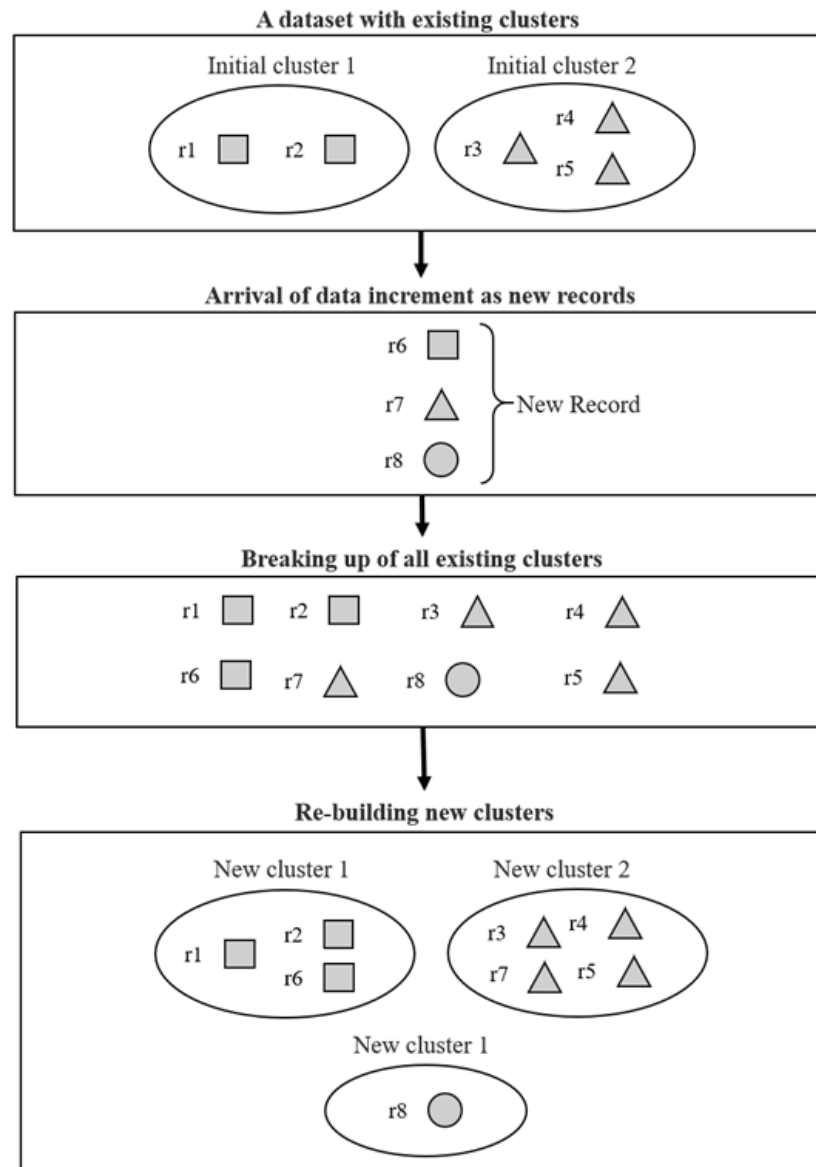


Figure 6.1: Stages of batch record linkage

Batch record linkage: Here, for each increment dataset ΔD , the record linkage process is executed for the whole dataset $D + \Delta D$. Let us assume a scenario that our base dataset contains one million records whereas our each increment contains one thousand records. In batch record linkage, we have one million as our starting point to apply clustering. When the first increment arrives, we have to perform clustering over one million and one thousand combined records. It is a time-consuming process hence inefficient. The process is illustrated in Figure 6.1.

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

linkage process.

Incremental record linkage (IRL): An incremental record linkage process preserves the clusters developed from the base dataset D and merges the records from the incremental dataset ΔD using a similarity function. IRL process creates new clusters if some of the records of ΔD is not similar to any of the existing clusters based on a similarity function. In order to get a practical understanding of how incremental record linkage works, Figure 6.2 will be helpful. Here, three boxes in the figure represent three distinct stages of the IRL process.

Definition 6.1 Incremental Record Linkage (IRL): : Let D be a set of records and ΔD be an increment to D . Let ρ_D be the clustering of records in D . Incremental record linkage clusters records in $D + \Delta D$ based on ρ_D . We denote the incremental record linkage method by f , and denote the results by $f(D, \Delta D, \rho_D)$.

The aim of incremental record linkage is to improve performance significantly compared to its corresponding batch linkage algorithm especially if the increment is small [67]. Specifically, the computation of $f(D, \Delta D, \rho_D)$ should be faster than the computation of $F(D + \Delta D)$ if $|\Delta D| \ll |D|$ holds. At the same time, incremental record linkage should achieve equivalent accuracy to its reference batch algorithm. We denote this constraint as $f(D, \Delta D, \rho_D) \approx F(D + \Delta D)$.

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

Definition 6.2 Privacy Preserving Incremental Record Linkage (PPiRL): Let D be a set of records and A is the set of attributes of D . Let \bar{A} is the set of sensitive attributes of D and $\bar{A} \subset A$. ΔD is an increment to D . We denote the privacy preservation method by Γ , and denote the privacy preserved results $\Gamma(D)$ by \bar{D} , and $\Gamma(\Delta D)$

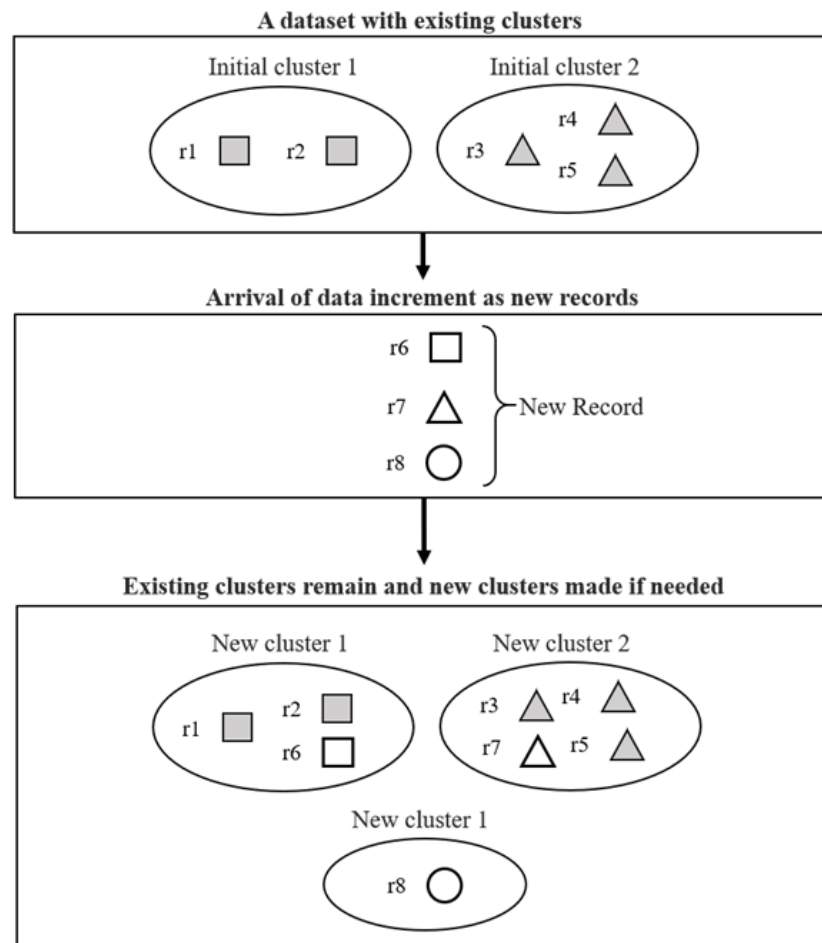


Figure 6.2: Stages of incremental record linkage

by $\bar{\Delta D}$ respectively. Let $\bar{\rho}_D$ be the clustering of records in \bar{D} . Privacy Preserving Incremental record linkage clusters records in $\bar{D} + \bar{\Delta D}$ based on $\bar{\rho}_D$. We denote the privacy preserved incremental record linkage method by f' , and denote the results by $f'(\bar{D}, \bar{\Delta D}, \bar{\rho}_D)$.

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

6.4 PPiRL, an End-to-End Framework

Our proposed solution is an end-to-end framework for record linkage which considers significant reduction of time for performing record linkage along with privacy preservation without compromising the linkage quality. There are five basic steps in the framework with different functionalities. Data pre-processing, privacy preservation, blocking, clustering, and evaluation are the five stages of our framework illustrated for base dataset in Figure 6.3 and increments in Figure 6.4.

6.4.1 Data Pre-processing

Pre-processing of data helps improve the condition of data by handling errors and inconsistencies from data. Although data quality issues are found in a single dataset, quality issues become serious when data is integrated from multiple sources into a warehouse [150]. Some essential steps of data pre-processing are feature selection, data standardization, data cleaning, missing data imputation, normalization, etc. More details of pre-processing is presented in Section 2.1.1.

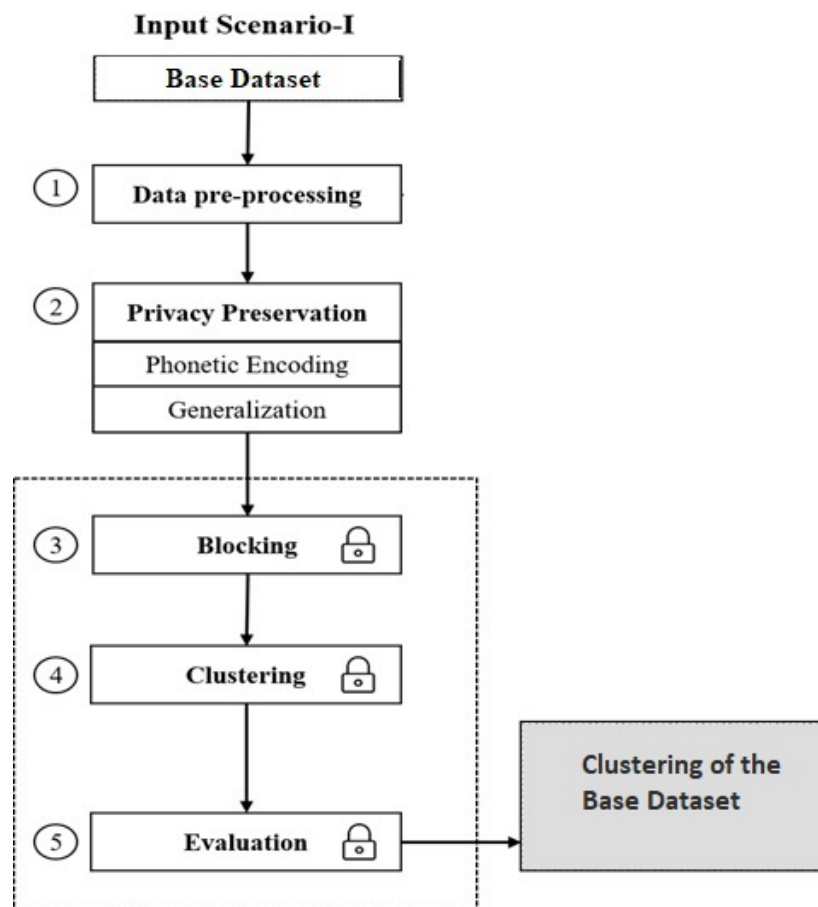


Figure 6.3: PPIRL, an end-to-end framework steps for the base dataset

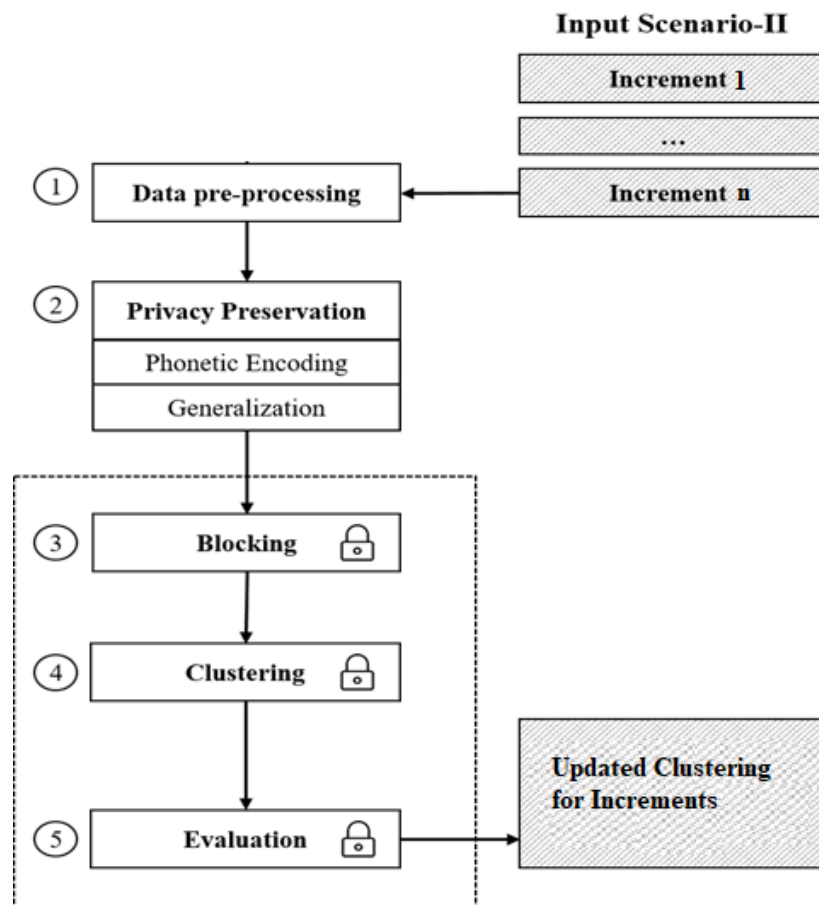


Figure 6.4: PPIRL, an end-to-end framework steps for increments

6.4.2 Privacy Preservation

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

6.4.2.1 Phonetic Encoding

The phonetic encoding algorithm groups values that have similar pronunciation. It inherently provides privacy and increases scalability as well. The procedure is illustrated in Figure 6.5. One of the most sensitive attributes in healthcare datasets is the name of the patients. A leak of a patient name could jeopardize the patient's privacy in a bad way. Using phonetic encoding we got the following advantages:

1. Names will be encoded. So they can not be easily identified.
2. Names will be generalized. That is similar pronouncing names with different spelling will produce same code.
3. Because of generalization of names, the output code is robust against noises and spelling errors, which are common in healthcare centers specially in the developing countries [104].

We used NameSig algorithm for phonetic encoding as it produces better results than Soundex, Metaphone and some other commonly used phonetic algorithms. Details of the algorithm can be found in Section 4.2.2.

6.4.2.2 K-anonymization Method

The k-anonymity is a popular generalization algorithm. The main purpose of generalization is to help overcome the problem that lies with record linkage which is re-identification of entities. Data generalization process generalizes data in a way that re-identifying the data to its source record is quite impossible. There are many generalization techniques. Among them, K-anonymization method has been proven to be an effective privacy technique which can preserve the privacy of record linkage

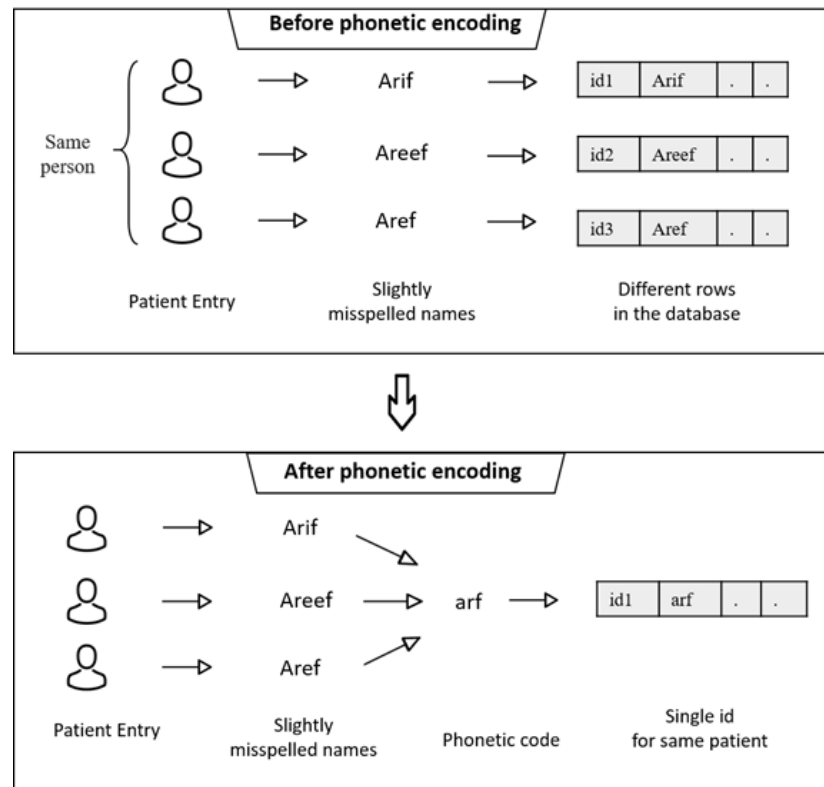


Figure 6.5: Process of phonetic encoding

results [16]. In K-anonymization technique, we assume that data related to a specific person is gathered in a dataset. The anonymization process is started by removing all the identifying features like SSN, explicit identifiers, etc. Even after removing the identifying features it is possible to find a person's data by finding a pattern in other features. To tackle that K-anonymization generalizes the feature values as much as possible with a value of K either being fixed at the beginning or getting an adaptive value as the linkage process continues. Figure 6.6 explains the k-anonymization process elaborately.

In Figure 6.6, we can see that common identifying feature such as name is suppressed at the very beginning. Then, other features which could be used to identify a person in a dataset is generalized by varying the value of K at a different time. In our case, we have used the adaptive value of K which allows us to select the best value of K depending on the results found.

Collected data

No.	Name	Post code	Sex	Age	Diagnosis
1	Jabir khan	1800005	Male	38	Cancer
2	Akib haider	1800012	Male	39	Cancer



Suppression

Anonymized data

No.	Name	Post code	Sex	Age	Diagnosis
1		1800005	Male	38	Cancer
2		1800012	Male	39	Cancer
3	suppressed	1800003	Male	37	Cancer
4		1810015	Female	40	HIV
5		1810015	Female	46	Cancer
6		1810013	Female	43	Measles



K-anonymization

K-anonymized data

No.	Post code	Sex	Age	Diagnosis
1	18000**	Male	30s	Cancer
2	18000**	Male	30s	Cancer
3	18000**	suppressed	30s	Cancer
4	18100**		40s	HIV
5	18100**	Female	40s	Cancer
6	18100**	Female	40s	Measles

} 3

} 3

Figure 6.6: Illustration of k-anonymization process

6.4.3 Blocking

For large datasets, comparing every record with all other records is computationally impractical. Blocking and indexing allows us to divide the whole datasets into blocks depending on some criteria. There are many existing techniques to achieve the task of blocking. Blocking based indexing groups similar record into one blocks. Only the records that are in the same block are compared in the comparison step to be classified as matches or non matches. Details of widely used blocking techniques are presented in Section 2.1.2.

For our framework, we have five identifiable attributes of the patient on which we have applied on our clustering algorithm. These attributes are Patient Name, Gender, Age, Contact Number, and Address. We used two blocking keys, Gender and Address, that significantly reduce the total number of comparison in the dataset. Both keys falls in the category of traditional blocking technique. Next we briefly describe about traditional blocking technique.

6.4.3.1 Traditional Blocking

Traditional blocking is a widely used indexing technique. In traditional blocking, one attribute or a combination of attributes is used for indexing or grouping similar records from a dataset. For example, if, in a dataset, "Postcode" is used as a blocking key, then, each generated block will contain only those records that have the same postcode. It helps to avoid comparison of all records in a dataset and reduces the comparison space. The attributes used for blocking are known as blocking keys (BK).

Real-world data contains errors and variations. We need to ensure that similar data will fall into the same blocks though they contain errors and variations. To solve this, attribute values can be converted into phonetic codes using encoding function. It helps to group similar data into the same block if they contain a typographical error. Popular phonetic algorithms such as Soundex, Metaphone and Double Metaphone are used to encode before blocking [41], [77].

The total number of records inserted into a block depends on the frequency distribution of blocking keys(BK). For example, If we use surname as blocking key, we will

get larger blocks for popular surnames such as Khan or Chowdhury. Large block sizes affect the efficiency and scalability of the record linkage process. To Avoid generating large block sizes Gu and Baxter [69] proposed an adaptive blocking technique which can solve from generating large block sizes.

Another issue with traditional blocking is the quality of the BK. If the value of selected features has too many missing values or error and variation, there is a possibility to get wrong blocks for those inserted records. It can affect the quality of the matching process. To overcome this issue, data preprocessing such as imputation of missing data and noise reduction are helpful.

In traditional blocking, comparisons are restricted to record pairs within each block. Blocking is generally implemented using sorting the two files on one or more variables. For example, if both files were sorted by zip code, the pairs to be compared would only be taken where zip codes agree. Record pairs disagreeing on zip code would not be considered and hence would be automatically classified as non-match. In Table 6.1 and Figure 6.7, it is illustrated how traditional blocking works using "Surname" as the blocking key.

Table 6.1: Use of surnames as blocking keys

Identifiers	Surnames	BK Values (Soundex)
R1	Smith	S530
R2	Miller	M460
R3	Peters	P362
R4	Myler	M460
R5	Smyth	S530
R6	Millar	M460
R7	Smyth	S530
R8	Miller	M460

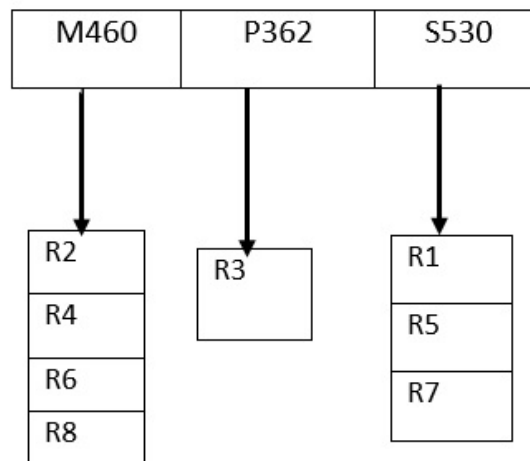


Figure 6.7: Traditional blocking using Soundex code

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

Pairs Completeness (*pc*): Pairs completeness measures the effectiveness of an indexing technique in the record linkage process. Pairs completeness is similar to the recall measure as used in information retrieval. Pairs Completeness It is calculated as $pc = B_M/N_M$.

Pairs Quality (*pq*): It measures the efficiency of an indexing technique. is similar to the precision measure as used in information retrieval. It is calculated as $pc = B_M/(B_M + N_M)$.

In Table 6.2, the performance of the two blocking keys that we have used for record linkage is presented. We have calculate *rr*, *pc* and *pq* for both Address and Gender attributes. It is clear from the table that, "Address" as the blocking key is better alternative than "Gender".

Table 6.2: Performance of the blocking keys

	Address as Blocking Key	Gender as Blocking Key
Time(s)	4.098	36.496
Reduction ratio	0.95	0.5
Pair Completeness	0.99	0.99
Pair Quality	0.97	0.28

6.4.4 Clustering

There are many clustering algorithms available. Because of the simple approach and easy implementation process compared to existing approaches we have chosen agglomerative hierarchical clustering and Davies–Bouldin (DB) index.

The principal of DB index is stated here. For each cluster C , similarities between C and all other clusters are computed first. Then the highest value is assigned to C as its cluster similarity value. Finally, the DB index can be obtained by averaging all the cluster similarities. The smaller the index value implies better clustering. By minimizing DB index, clusters will be more distinct from each other and best partitioning can be obtained. DB index was primarily defined for Euclidean space. So we need some adjustment in the definition of distance to apply it for record linkage purposes. We adopt the definition described in [67] as follows. For each cluster C , the intra-cluster distance is defined as the complement of average similarity between records in the cluster; that is,

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

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

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

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

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

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

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

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

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

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

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

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

6.4.5 Evaluation

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

6.4.5.1 Linkage Evaluation

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

6.4.5.2 Privacy Evaluation

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

6.5 Experimental Results

In this section, we present the results of different experiments based on the real-world and synthetic datasets. Here, we have used the steps of our framework of Section 6.4. Our algorithm is presented next.

Algorithm 6.1: PPIRL

```

1 Input: Recordset for Record Linkage (Base dataset or an Increment);
2 Output: Record Linkage results;
3 while !End of Recordset do
4   Preprocess recordset;
5   Feature Selection;
6   Standardization;
7   Phonetic Encoding of Text attributes;
8   K-Anonymization of Numeric attributes;
9   Apply clustering;
10 end

```

6.5.1 Data Pre-processing

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

6.5.1.1 Feature Selection

Feature selection is a procedure where a subset of original features is selected by following some criteria. To select necessary features for our framework, we have followed the basic steps of feature selection. First, we have collected all features from the recordset to be linked. Second, we have generated a candidate set, a subset of the whole set, which contained some selected features from the dataset using chi-square test and with the help of domain knowledge. Finally we have found a set of five features. They are the patient name, gender, age, contact number, and address. Table 6.3 lists the selected features via this process. In the left column of Table 6.3, we can see fifteen features which are fundamental features in the raw dataset and in the right column we can see the finally selected five features. Details of feature selection is out

of scope of this chapter.

Table 6.3: Feature selection

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

6.5.1.2 Normalization of Age Values

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

uniform age unit is needed for data mining task. We applied normalization techniques to transform the age to hold only age in year format.

Table 6.4: Age normalization

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

In Table 6.4, we can see the actual age values that appear in the raw dataset such as 7 Y 4 M. This type of changing values in units is harmful for our calculation. So, we have transformed the age values to a standard from which can be seen in the right most column of Table 6.4. We can see the transformed age values as they all appear with the year as their unit of representation. This helps the calculation of the Age feature in the patient dataset.

6.5.1.3 Address Standardization

The addresses provided by the patients in healthcare data is also very noisy and unstructured. For that reason, we have come up with the idea of extracting address in strict format and followed the standard provided by the Bangladesh Bureau of Statistics (BBS) of the Ministry of Planning. BBS provides a GEO code list up to Upazila label of Bangladesh. So, we extracted the raw addresses and formatted them as the desired format of BSS. Then we applied the GEO codes from the code list with the help of our algorithm. Below we can see both the extracted addresses and the mapped geocodes for the corresponding addresses. In Table 6.5, the geocodes are shown as they are formed.

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

Table 6.5: Mapping address to geocode

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

6.5.2 Experimental Setup

Working environment: We used a PC with Intel Core i7 CPU of 2.40 GHz processing speed and 8GB RAM, run on 64-bit Windows 10 operating system for the experiments.

Implementations: To determine the effectiveness of our framework, we implemented the following algorithms:

- nameGist, the phonetic encoding algorithm, groups the similar sounding names together and giving privacy to the ‘Name’ feature as well.
- K-Anonymization, the privacy-preserving algorithm, ensures the generalization of ‘Contact,’ ‘Address’ features.
- NAIIVE, the incremental baseline algorithm, compares each inserted record with existing clusters, then either add it into an existing cluster or creates a new cluster for it.
- Correlation Clustering applies correlation penalty to get the best cluster results

while implementing clustering.

6.5.3 Linkage Evaluation

6.5.3.1 External validation Measure Results

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

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

For privacy, we considered the frequency distribution of various sensitive features in the dataset. We have applied two types of a clustering algorithm for our incremental record linkage application. One of the algorithms is Naive incremental, and the other one is correlation clustering. We applied the algorithms on our dataset with varying noise. We introduced intentional noise in our dataset ranging from 5% to 10% of the total records in the dataset. The results can be understood in two aspects, one is accuracy, and the other is time efficiency. Both these two aspects give us the overall performance of a blocking key. We have used geocode as blocking key. After using different attributes as blocking key, we got better performance by taking geocode as the blocking key.

We also implemented the correlation clustering in order to compare with an existing solutions. Table 6.6 indicates the results we received for Correlation and Naive clustering using the geocode as the only blocking key. For various noise percentage of the dataset, we have calculated the precision, recall, and F-measure of the linkage results. As Correlation clustering checks all the pairs between two comparing clusters iteratively we can see that for even no noise in the dataset, it takes more time.

Table 6.6: PPIRL performance on real data with various noise setting

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

We have also tested the performance of our PPIRL framework using a synthetic dataset generated using Python script. The attributes of the dataset are identical to our real health dataset. We used geocode as the blocking key for this dataset. Initially it the base dataset contains one thousand records. We have also generate a total of seventy-five records for three increment (twenty-five records for each). We have used geocode as the blocking key. The performance of PPIRL over the synthetic dataset is presented in the Table 6.7.

Table 6.7: Performance of PPIRL using synthetic dataset

Dataset Description	F-measure	Time
Initial Dataset	0.974	4.58
Increment-1	0.966	0.07
Increment-2	0.963	0.16
Increment-3	0.964	0.08

6.5.3.2 Internal Validation Measure Results

To evaluate the linkage results we have also calculated internal evaluation matrices. We have used correlation penalty and DB-Index penalty particularly to evaluate the linkage quality. The lesser the penalty, the better the results. In Table 6.8 we have shown the correlation penalty and DB-Index penalty for Naive and Correlation clustering. The equation for penalty is presented in Section 6.4.4.

Table 6.8: Penalty evaluation for naive and correlation clustering

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

In Table 6.8, we can see that the Correlation penalty is much higher than DB-Index penalty for Naive clustering. As DB-Index uses more robust formula, we can say the performance of our algorithm is better in this regard. We have also shown that the correlation penalty and DB-Index penalty for Correlation clustering for linkage purposes on our dataset.

6.5.4 Privacy Evaluation

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

6.5.4.1 Frequency Analysis

Frequency distribution of the characters of certain attributes in a dataset may cause information exposure. In our experimental dataset, there are several identifying features of a patient. Among them, the ‘Name’ feature is a sensitive one. To get a hold of this feature one way is to get the frequency distribution of English letters occurring in the names. If the frequency of letters remains same even after applying privacy techniques to encode the names, then it is quite possible for the names to be at the hands of an unwanted outsider or in the worst case a hacker using frequency distribution. Figure 6.8 is the representation of the frequency of letters found in the original names before any privacy algorithm is applied.

The figure is a histogram representation of the frequency of the letters. It can be

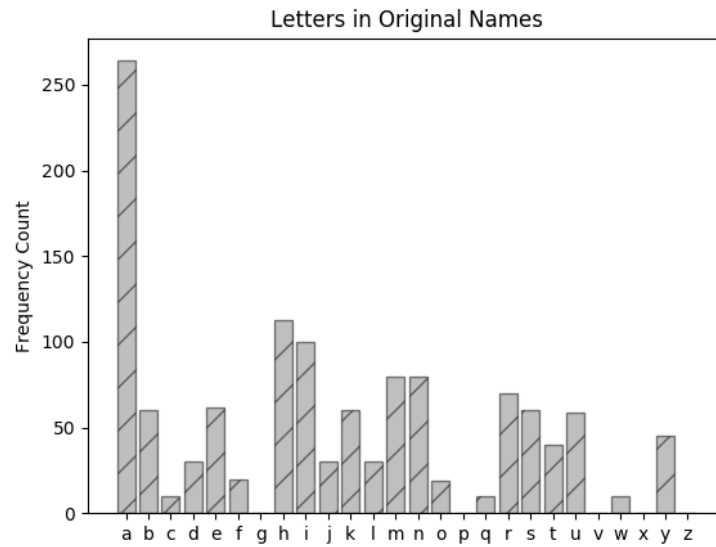


Figure 6.8: Frequency analysis of original 'Name' attribute

deducted from the graph that certain letters appear more than others making the graph a skewed graph. To check the authenticity of the privacy algorithm that it works on the Name feature we have a type of phonetic encoding algorithm named nameGist which encodes the name of the patient into a code depending on the phonetic characteristics of the name. We have illustrated the frequency of letters in the names of the patient after the application of this nameGist algorithm.

In Figure 6.9, we can see the frequency of each letter found in the encoded version of the Name feature of the patient dataset. The figure is a graph representation of the frequency of the letters. This graph is significantly different from the graph that is showed in Figure 6.8 because there was only one letter which had a frequency higher than 100, but in this graph, we can see that more than two letters exceed the label of 100. Also, after more close inspection we can say that the frequency of this graph different from the last one and has a more well-balanced frequency of the letters. This balance is crucial as the hackers rely on the similarity of frequency to steal the valuable information from health datasets. The difference of frequency count is visible. It is also notable that frequency count has decreased significantly in the second figure which makes the privacy of the Name attribute of the patients quite safe. Thus, if a hacker even gets hold of the encrypted dataset, it is quite impossible to get the actual

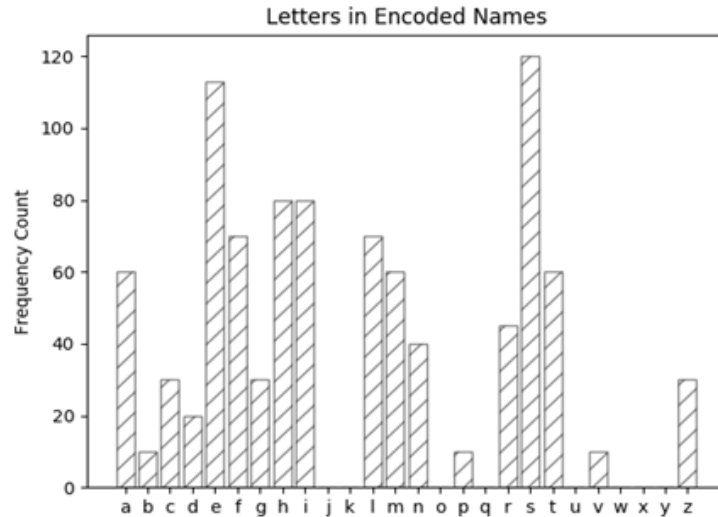


Figure 6.9: Frequency analysis of 'Name' attribute after privacy enforcement

names of the patients from it.

6.5.4.2 Dictionary Attack

A dictionary attack is usually carried out to break passwords or similar encrypted data in a database with the help of an available digital dictionary [133]. To carry out a successful dictionary attack, a hacker must have access to a dictionary or list of frequently used words or vocabularies. General dictionaries are useful in this matter as they provide millions of words which could be used to create a password for a user. Our dataset does not contain passwords, but it contains encoded names.

As a dictionary contains so many words that normally people's usable password contain in there and hackers used a technique to go through all the word on dictionary and find a word or something to encrypt the data. Patients name encoded by our system can be marked as safe from dictionary attacks for two reasons. First, most names are not available in dictionaries. Second, as we performed phonetic encoding, the output encoded names are generalized.

6.5.4.3 Information Gain

By simulating the framework under different adversary models, we can evaluate the proof of privacy of PPIRL. The less information is extracted from the framework, the

safer it is. Here we have considered popular Honest-but-curious (HBC) Adversary Model. In HBC model, each of the parties is obliged to follow the protocol. Here, a party does not forget the knowledge that it learns during the information exchange. In other words, all parties are curious, in the sense that they try to find out information of other parties, as much as possible, despite of following the protocol [143]. In the perspective of HBC, a protocol is secure if and only if all parties involved have gained no new knowledge at the end of the exchange other than what they would have learned from the output of the record pairs classified as matches. Most of the PPRL solutions proposed in the literature assume the HBC adversary model [183].

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

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

Low entropy means low uncertainty as a result higher predictability. In Table 6.9 calculated value of entropy for the sensitive attributes and concatenated value is presented. Here we calculated entropy for bit per character in a message. We can see from the table that concatenated attribute's entropy is higher than that of each individual attribute thus producing lower predictability of real data.

Table 6.9: Entropy of individual attributes and concatenated data

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

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

equation is given below:

$$H(Y|X) = \sum_j p(X = v)H(Y|X = v)$$

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

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

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

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

Table 6.10: Information gain

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

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

6.5.5 Comparison of PPiRL with Batch Record Linkage

Batch record linkage process considers all the new records and does not maintain the cluster from the last linkage when they arrive. On the other side, PPiRL keeps track of the cluster of the last linkage. So, when new updates arrive, it performs faster than Batch linkage. Table 8 briefly illustrates the information between these two approaches.

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

6.5.6 Comparison of PPiRL with Incremental Record Linkage

We have also compared PPiRL with IRL. Privacy preservation techniques were not used in IRL so we have an extra advantage there already as we can see in table 9. We have gotten privacy assurance of 81% in our framework. Although our framework got little less in linkage quality, it was tradeoff due to privacy techniques which are considerable in this type of sensitive researches.

Table 6.11: Comparison between PPiRL and batch record linkage

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

6.6 Summary

This chapter proposes an end-to-end framework that conducts privacy-preserving algorithms as well as record linkage algorithms in an incremental fashion. Our algorithms ensure the privacy of the sensitive records and also maintain the linkage of the records by creating a proper cluster of similar records. Being the first to experiment in this

Table 6.12: Comparison between PPiRL and IRL

Feature	Technique used for RL	
	IRL	PPiRL
Privacy Preservation Technique	None	Phonetic encoding & Generalization
Information gain by other party	Full	19%
Linkage quality	95%	91%

field, we could only apply very few algorithms to test our framework. Combining privacy with incremental record linkage has paved the way to secure linkage of sensitive data residing in several public and private organizations meeting the demand of big data era. Experimental results with 65000 records from multiple datasets show that our framework can achieve around 90% correct record linkage with much reduced times. Different privacy attack, executed by external body, showed that our framework is stable against well known attacks e.g., dictionary attack, frequency attack. The information gain from the exchange record using HBC model is also less than 20%. In future, we want to improve the linkage quality, privacy and performance of the framework using other state of the art algorithms.

Chapter 7

Conclusions

We conclude the thesis by briefly describing the research problems we have addressed and our contributions. We also outline possible future directions in this area of research. In this thesis, we have focused on privacy-preserving record linkage (PPRL) techniques. To perform PPRL efficiently in the presence of noise, we have proposed new algorithms for missing value imputation, phonetic encoding, and key-based record linkage.

In the big-data era, traditional (batch) record linkage faces two challenges: high velocity of data updates and massive database sizes. These challenges require an incremental solution so that when data updates appear, record linkage results can be updated swiftly. We are the first to propose an incremental privacy-preserving record linkage technique (PPiRL). We have proposed a novel PPiRL framework for performing record linkage efficiently, maintaining privacy, and scalability. The findings of this thesis are presented in the following sections.

7.1 Missing Data Imputation

Missing data create problems in the record linkage process as similarity of records are used for linkage. When an attribute's value is missing for some tuples, similarity can not be calculated correctly. Incorrect imputation of missing values could lead to erroneous record linkage results.

For real-world health datasets, missing data is a common phenomenon. We have

studied widely used algorithms for missing data imputation. There are two categories of imputation methods: single and multiple imputation. Single imputations are easier to implement but may produce biased imputation for a missing data. Multiple imputation can overcome the bias at the cost of complexity to use.

We have implemented twelve imputation algorithms of both categories for three types of attributes: binary, ordinal, and numeric. We have used the MICE package in R to implement multiple imputation algorithms. To achieve better accuracy, precision, recall, and F-measure, we have provided an extension of the MICE package. We named it "SICE" which performs better than MICE and several other imputation methods for numeric and binary attributes. Details have been described in Chapter 3.

7.2 Phonetic Encoding

Phonetic encoding by using different phonetic algorithms is necessary for many applications, including name-matching, database record linkage, noise reduction, search recommendations, etc. A phonetic algorithm can withstand an incorrect spelling of a name by generating the same code, which helps to solve the problem of identifying all records of a person during record linkage. Phonetic algorithms support the privacy-preserving record linkage process in two ways. First, by reducing noise, these algorithms help to improve linkage accuracy. Second, they have an inherent privacy preservation characteristic that is used by many privacy-preserving record linkage techniques.

A common problem with health datasets is the presence of different types of noise. Some reasons are typographical error, hardware problem, human error, etc. Phonetic algorithms help to reduce noises from names and other strings. We have studied widely used algorithms for English and Bengali phonetics, e.g., Soundex, NYSIIS, Metaphone, NameValue, etc. We have proposed a novel phonetic algorithm nameGist, which is the only algorithm that supports both English and Bengali name matching. Our proposed algorithm performs significantly better than existing Bengali phonetic algorithms and also can efficiently process English phonetic names. Details have been described in

Chapter 4.

7.3 Key Based Privacy Preserving Record Linkage

Record linkage is a useful task in many application areas, e.g., healthcare, finance, census, etc. Linking records is an easy task in the presence of universal IDs such as SSN or national ID number. However, in most of the cases, these universal ID numbers are not available in the databases that we wish to link. For such cases, quasi-identifiers (QID), i.e., name, address, gender, etc., can be used for record linkage. Privacy-preserving record linkage (PPRL) using QIDs is a challenging task itself, and even, more challenge is added for the health dataset of the developing countries like Bangladesh. This is due to the nature of additional noise found in the health dataset of the developing countries.

We have proposed an improved PPRL technique, Key-based Secured Record Linkage (KSRL), for the constrained health datasets. We have categorized the patient identifiable attributes into three categories: changeable attributes, fixed unambiguous attributes, and fixed ambiguous attributes. Empirical results show that KSRL can effectively connect records in the absence of universal ID numbers and the presence of erroneous data, e.g., misspelled of patient Name.

7.4 Incremental Privacy Preserving Record Linkage

Privacy-preserving record linkage faces two challenges at the edge of big data. First, the high velocity of data updates swiftly makes previous linkage results extinct. Second, a massive volume of data requires extensive time for applying record linkage. These two challenges require an incremental solution so that when data updates appear, we can swiftly update linkage results. We are the first one to recognize Privacy-Preserving Incremental Record Linkage as a new field of research. Recognition of this field paves the way for solving the problems of data mining relating to volume and velocity of data along with privacy issues. We have proposed a new end-to-end framework that encompasses both the privacy and linkage of data. It can be derived from our

experiments that it is possible to maintain privacy while applying incremental updates in record linkage applications.

We have proposed an end-to-end framework that conducts privacy-preserving algorithms as well as record linkage algorithms in an incremental fashion when updates of the data arrive. Our algorithms not only ensure the privacy of the sensitive patient records of a medical dataset but also successfully maintain the linkage of the records by creating a proper cluster of similar patients. The results found through the experiments prove that our framework is suitable for the privacy-preserving incremental record linkage. We have achieved almost similar accuracy as batch-PPRL techniques with much faster processing time. We have also tested the ability of privacy preservation of our developed framework using different privacy attacks, e.g., dictionary attack, frequency analysis, and information gain, and found satisfactory results. Combining privacy preservation with incremental record linkage has paved the way for a secure linkage of a large amount of sensitive data residing in numerous public and private organizations.

7.5 Broader Impacts of the Thesis

The results of this thesis will advance knowledge and understanding in the general area of record linkage of sensitive healthcare data. This understanding will be significant for the development and enhancement of national health data warehouses of Bangladesh or other countries. The knowledge discovered by mining healthcare warehouse data could be used for the development of better healthcare services of Bangladesh. Supporting Online Analytical Processing (OLAP), the warehouse will help the healthcare decision-makers to develop better policies in the area of healthcare systems. As this research has been supported by the ICT Fellowship of the Government of Bangladesh, the Government could take the initiative to utilize the knowledge gathered from this research to improve the health information systems of Bangladesh. Advances from the research topics will be disseminated widely through publications in reputed academic journals and conferences. The philosophy of this research is "Analyzing health data

for better healthcare"!

7.6 Future Works

The research presented in this thesis opens many avenues that could be explored in the future. They are summarized as follows.

- A missing data framework is to be developed, which will work as a generalized solution of missing data and will be applicable for any kind of attributes and any kind of datasets. Especially SICE should be extended for better performance in the case of nominal and ordinal data.
- Improvement of our developed phonetic algorithm nameGist can be made by further reducing the false positive rate. For Bengali name encoding, time should be reduced using better pre-processing techniques. More language support needs to be added for better performance in the global arena, e.g., Chinese, Hindi, and Arabic languages.
- For the PPiRL framework, new privacy preservation algorithms such as bloom filters can be introduced to reduce the current information gain of PPiRL. Different blocking and comparison techniques may also be tested in this framework to improve accuracy.
- Record linkage applications facilitate the process of data mining from the matched records of multiple databases. Different machine learning algorithms may be applied to the results of PPiRL to discover hidden knowledge and interesting patterns.

Finally, we will present the list of published papers from this thesis in different conferences and journals. We have also included the papers submitted to the journals for publication in the list, with the status "submitted."

Publications

The research conducted in this thesis has resulted in the following publications.

1. Shahidul Islam Khan and Abu Sayed Md. Latiful Hoque, "Incremental record linkage with privacy preservation." submitted to the VLDB Journal, Springer.
2. Shahidul Islam Khan and Abu Sayed Md. Latiful Hoque, "SICE: An improved missing data imputation technique." Journal of Big Data, Springer, Vol. 7, No. 1, 2020, pp.1-21.
3. Shahidul Islam Khan and Abu Sayed Md. Latiful Hoque, "Secured technique for healthcare record linkage." in Proceedings of the 6th International Conference on Networking, Systems and Security (NSysS-2019), 2019, Full Paper, Published by ACM.
4. Shahidul Islam Khan, Md Mahmudul Hasan, Mohammad Imran Hossain, and Abu Sayed Md Latiful Hoque, "nameGist: a novel phonetic algorithm with bilingual support." International Journal of Speech Technology, Springer, Vol. 22, No. 4, 2019, pp. 1135-1148.
5. Shahidul Islam Khan and Abu Sayed Md. Latiful Hoque, "Health Data Integration with Secured Record Linkage: A Practical Solution for Bangladesh and Other Developing Countries." in Proceedings of 3rd International Conference Networking, Systems and Security (NSYSS-2017), 2017, Published by IEEE.
6. Shahidul Islam Khan and Abu Sayed Md. Latiful Hoque, "Similarity analysis of patients' data: Bangladesh perspective." in Proceedings of International Conference on Medical Engineering, Health Informatics and Technology (MediTec-2016),

2016, Published by IEEE.

7. Shahidul Islam Khan and Abu Sayed Md. Latiful Hoque, "Privacy and security problems of national health data warehouse: A convenient solution for developing countries." in Proceedings of 2nd International Conference on Networking Systems and Security (NSysS 2016), pp. 157-162, 2016, Published by IEEE.
8. Shahidul Islam Khan, Abu Sayed Md. Latiful Hoque and Mohammad Ullah, "National health data warehouse Bangladesh for remote health monitoring: features, problems and privacy issues." in Proceedings of Remote Health Monitoring Workshop (rHealth 2016), 2016, Published by Dept. of CSE, BUET.
9. Shahidul Islam Khan and Abu Sayed Md. Latiful Hoque, "An analysis of the problems for health data integration in Bangladesh." in Proceedings of International Conference on Innovations in Science, Engineering and Technology (ICISSET-2016), 2016, Published by IEEE.
10. Shahidul Islam Khan and Abu Sayed Md. Latiful Hoque, "Digital health data: A comprehensive review of privacy and security risks and some recommendations." Computer Science Journal of Moldova, Vol. 24, No. 2, pp. 273-292, 2016, Published by the Academy of Science, Moldova.
11. Shahidul Islam Khan and Abu Sayed Md. Latiful Hoque, "Towards development of national health data warehouse for knowledge discovery", Advances in Intelligent Systems and Computing, Vol. 385/2, pp.413-421, 2015, Springer.

Bibliography

- [1] Noha Adly. Efficient record linkage using a double embedding scheme. In DMIN, pages 274--281, 2009.
- [2] Soon Ae Chun and Bonnie MacKellar. Social health data integration using semantic web. In Proceedings of the 27th annual ACM symposium on applied computing, pages 392--397, 2012.
- [3] A Al-Kadi Ibrahim. The origins of cryptology: The arab contributions, 1992.
- [4] Ali Al-Lawati, Dongwon Lee, and Patrick McDaniel. Blocking-aware private record linkage. In Proceedings of the 2nd international workshop on Information quality in information systems, pages 59--68. ACM, 2005.
- [5] Mohammad Allahbakhsh, Aleksandar Ignjatovic, Boualem Benatallah, Elisa Bertino, Norman Foo, et al. Collusion detection in online rating systems. In Asia-Pacific Web Conference, pages 196--207. Springer, 2013.
- [6] Sarmad Alshawi, Farouk Missi, and Tillal Eldabi. Healthcare information management: the integration of patients' data. Logistics Information Management, 2003.
- [7] Mehran Amiri and Richard Jensen. Missing data imputation using fuzzy-rough methods. Neurocomputing, 205:152--164, 2016.
- [8] Arvind Arasu, Michaela Goetz, and Raghav Kaushik. On active learning of record matching packages. In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, pages 783--794, 2010.

- [9] Raghunath Arnab. Survey sampling theory and applications. Academic Press, 2017.
- [10] Yonatan Aumann and Yehuda Lindell. Security against covert adversaries: Efficient protocols for realistic adversaries. *Journal of Cryptology*, 23(2):281--343, 2010.
- [11] Tobias Bachteler, Rainer Schnell, and Jörg Reiher. An empirical comparison of approaches to approximate string matching in private record linkage. In *Proceedings of Statistics Canada Symposium*, volume 2010. Citeseer, 2010.
- [12] Dixie B Baker, Bartha M Knoppers, Mark Phillips, David van Enckevort, Petra Kaufmann, Hanns Lochmuller, and Domenica Taruscio. Privacy-preserving linkage of genomic and clinical data sets. *IEEE/ACM transactions on computational biology and bioinformatics*, 16(4):1342--1348, 2018.
- [13] Suresh Balakrishnama and Aravind Ganapathiraju. Linear discriminant analysis-a brief tutorial. *Institute for Signal and information Processing*, 18:1--8, 1998.
- [14] Carlo Batini, Monica Scannapieco, et al. *Data and information quality*. Cham, Switzerland: Springer International Publishing. Google Scholar, 2016.
- [15] Rohan Baxter, Peter Christen, Tim Churches, et al. A comparison of fast blocking methods for record linkage. In *ACM SIGKDD*, volume 3, pages 25--27. Citeseer, 2003.
- [16] Roberto J Bayardo and Rakesh Agrawal. Data privacy through optimal k-anonymization. In *21st International conference on data engineering (ICDE'05)*, pages 217--228. IEEE, 2005.
- [17] Zohra Bellahsene, Angela Bonifati, and Erhard Rahm. *Schema matching and mapping*. Springer, 2011.
- [18] Matthew J Bietz, Cinnamon S Bloss, Scout Calvert, Job G Godino, Judith Gregory, Michael P Claffey, Jerry Sheehan, and Kevin Patrick. *Opportunities and challenges*

- in the use of personal health data for health research. *Journal of the American Medical Informatics Association*, 23(e1):e42--e48, 2016.
- [19] Mikhail Bilenko and Raymond J Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 39--48, 2003.
- [20] Jens Bleiholder and Felix Naumann. Data fusion. *ACM Computing Surveys (CSUR)*, 41(1):1, 2009.
- [21] Burton H Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422--426, 1970.
- [22] Dankmar Böhning. Multinomial logistic regression algorithm. *Annals of the institute of Statistical Mathematics*, 44(1):197--200, 1992.
- [23] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144--152, 1992.
- [24] Amiangshu Bosu, Fang Liu, Danfeng Yao, and Gang Wang. Collusive data leak and more: Large-scale threat analysis of inter-app communications. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 71--85, 2017.
- [25] M. L. Braunstein. *Practitioner's Guide to Health Informatics*. Springer, 2015.
- [26] Data breach today. Why hackers are targeting health data. <http://www.databreachtoday.asia/hackers-are-targeting-health-data-a-7024>.
- [27] J Michael Brick and Graham Kalton. Handling missing data in survey research. *Statistical methods in medical research*, 5(3):215--238, 1996.
- [28] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann. The balanced accuracy and its posterior distribution. In *2010 20th International Conference on Pattern Recognition*, pages 3121--3124. IEEE, 2010.

- [29] Adrian P Brown, Anna M Ferrante, Sean M Randall, James H Boyd, and James B Semmens. Ensuring privacy when integrating patient-based datasets: new methods and developments in record linkage. *Frontiers in public health*, 5:34, 2017.
- [30] Marvin L Brown and John F Kros. Data mining and the impact of missing data. *Industrial Management & Data Systems*, 2003.
- [31] Stefan Burkhardt, Andreas Crauser, Paolo Ferragina, Hans-Peter Lenhof, Eric Rivals, and Martin Vingron. q-gram based database searching using a suffix array (quasar). In *Proceedings of the third annual international conference on Computational molecular biology*, pages 77--83, 1999.
- [32] S van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of statistical software*, pages 1--68, 2010.
- [33] National Statistical Office Canada. Age categories, life cycle groupings. <https://www.statcan.gc.ca/eng/concepts/definitions/age2>.
- [34] Modern Health Care. Lincare ordered to pay 239,800 hipaa privacy penalty. <http://www.modernhealthcare.com/article/20160209/NEWS/160209856/lincare-ordered-to-pay-239800-hipaa-privacy-penalty>.
- [35] Ismael Castillo, Johannes Schmidt-Hieber, Aad Van der Vaart, et al. Bayesian linear regression with sparse priors. *The Annals of Statistics*, 43(5):1986--2018, 2015.
- [36] CMDS CEU. Reported breaches of compromised personal records in europe. <http://cmds.ceu.edu/sites/cmcs.ceu.hu/files/attachment/article/663/databreachesineurope.pdf>.
- [37] Daniel Chang. Jackson health: 'rogue' employee suspected of stealing private patient information. <http://www.miamiherald.com/news/health-care/article59339038.html>.

- [38] Moses Charikar, Chandra Chekuri, Tomàs Feder, and Rajeev Motwani. Incremental clustering and dynamic information retrieval. *SIAM Journal on Computing*, 33(6):1417--1440, 2004.
- [39] Peter Christen. A comparison of personal name matching: Techniques and practical issues. In *Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on*, pages 290--294. IEEE, 2006.
- [40] Peter Christen. Automatic record linkage using seeded nearest neighbour and support vector machine classification. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 151--159, 2008.
- [41] Peter Christen. *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection*. Springer Science & Business Media, 2012.
- [42] Peter Christen. A survey of indexing techniques for scalable record linkage and deduplication. *IEEE transactions on knowledge and data engineering*, 24(9):1537--1555, 2012.
- [43] Peter Christen and Karl Goiser. Quality and complexity measures for data linkage and deduplication. In *Quality measures in data mining*, pages 127--151. Springer, 2007.
- [44] Peter Christen, Rainer Schnell, Dinusha Vatsalan, and Thilina Ranbaduge. Efficient cryptanalysis of bloom filters for privacy-preserving record linkage. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 628--640. Springer, 2017.
- [45] Tim Churches and Peter Christen. Some methods for blindfolded record linkage. *BMC Medical Informatics and Decision Making*, 4(1):9, 2004.
- [46] Krzysztof J Cios and G William Moore. Uniqueness of medical data mining. *Artificial intelligence in medicine*, 26(1-2):1--24, 2002.

- [47] Chris Clifton, Murat Kantarcioğlu, AnHai Doan, Gunther Schadow, Jaideep Vaidya, Ahmed Elmagarmid, and Dan Suciu. Privacy-preserving data integration and sharing. In Proceedings of the 9th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, pages 19--26, 2004.
- [48] M Cochinwala, S Dalal, Ahmed K Elmagarmid, and VS Verykios. Record matching: Past, present and future. 2001.
- [49] William W Cohen, Pradeep Ravikumar, Stephen E Fienberg, et al. A comparison of string distance metrics for name-matching tasks. In IWeb, volume 2003, pages 73--78, 2003.
- [50] William W Cohen and Jacob Richman. Learning to match and cluster large high-dimensional data sets for data integration. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 475--480. ACM, 2002.
- [51] Joseph Conn. Hospital pays hackers 17,000 to unlock ehers frozen in 'ransomware' attack. <http://www.modernhealthcare.com/article/20160217/NEWS/160219920/hospital-pays-hackers-17000-to-unlock-ehers-frozen-in-ransomware>.
- [52] David De Brou and Mark Olsen. The guth algorithm and the nominal record linkage of multi-ethnic populations. *Historical Methods: A Journal of Quantitative and Interdisciplinary History*, 19(1):20--24, 1986.
- [53] Moniek CM de Goeij, Merel van Diepen, Kitty J Jager, Giovanni Tripepi, Carmine Zoccali, and Friedo W Dekker. Multiple imputation: dealing with missing data. *Nephrology Dialysis Transplantation*, 28(10):2415--2420, 2013.
- [54] Timothy De Vries, Hui Ke, Sanjay Chawla, and Peter Christen. Robust record linkage blocking using suffix arrays and bloom filters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(2):1--27, 2011.
- [55] Dimas Cassimiro do Nascimento, Carlos Eduardo Santos Pires, and Demetrio Gomes Mestre. Heuristic-based approaches for speeding up incremental record linkage. *Journal of Systems and Software*, 137:335--354, 2018.

- [56] Elizabeth A Durham, Murat Kantarcioglu, Yuan Xue, Csaba Toth, Mehmet Kuzu, and Bradley Malin. Composite bloom filters for secure record linkage. *IEEE transactions on knowledge and data engineering*, 26(12):2956--2968, 2013.
- [57] L Dusserre, C Quantin, and H Bouzelat. A one way public key cryptosystem for the linkage of nominal files in epidemiological studies. *Medinfo*, 8:644--647, 1995.
- [58] Ahmed K Elmagarmid, Panagiotis G Ipeirotis, and Vassilios S Verykios. Duplicate record detection: A survey. *IEEE Transactions on knowledge and data engineering*, 19(1):1--16, 2006.
- [59] Vladimir Estivill-Castro and Ahmed HajYasien. Fast private association rule mining by a protocol for securely sharing distributed data. In *2007 IEEE Intelligence and Security Informatics*, pages 324--330. IEEE, 2007.
- [60] Jianqing Fan, Fang Han, and Han Liu. Challenges of big data analysis. *National science review*, 1(2):293--314, 2014.
- [61] Ivan P Fellegi and Alan B Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183--1210, 1969.
- [62] Frank Franzak, Dennis Pitta, and Steve Fritsche. Online relationships and the consumer's right to privacy. *Journal of Consumer marketing*, 18(7):631--642, 2001.
- [63] Zhichun Fu, Peter Christen, and Mac Boot. Automatic cleaning and linking of historical census data using household information. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 413--420. IEEE, 2011.
- [64] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *Vldb*, volume 99, pages 518--529, 1999.
- [65] John W Graham, Patricio E Cumsille, and Allison E Shevock. Methods for handling missing data. *Handbook of Psychology, Second Edition*, 2, 2012.
- [66] Shaun J Grannis, J Marc Overhage, and Clement J McDonald. Analysis of identifier performance using a deterministic linkage algorithm. In *Proceedings of the AMIA Symposium*, page 305. American Medical Informatics Association, 2002.

- [67] Anja Gruenheid, Xin Luna Dong, and Divesh Srivastava. Incremental record linkage. *Proceedings of the VLDB Endowment*, 7(9):697--708, 2014.
- [68] Jerzy W Grzymala-Busse and Witold J Grzymala-Busse. Handling missing attribute values. In *Data mining and knowledge discovery handbook*, pages 33--51. Springer, 2009.
- [69] Lifang Gu and Rohan Baxter. Adaptive filtering for efficient record linkage. In *Proceedings of the 2004 SIAM International Conference on Data Mining*, pages 477--481. SIAM, 2004.
- [70] Lifang Gu and Rohan Baxter. Decision models for record linkage. In *Data mining*, pages 146--160. Springer, 2006.
- [71] Lifang Gu, Rohan Baxter, Deanne Vickers, and Chris Rainsford. Record linkage: Current practice and future directions. *CSIRO Mathematical and Information Sciences Technical Report*, 3:83, 2003.
- [72] Sara Hajian, Josep Domingo-Ferrer, and Oriol Farràs. Generalization-based privacy preservation and discrimination prevention in data publishing and mining. *Data Mining and Knowledge Discovery*, 28(5-6):1158--1188, 2014.
- [73] Alon Halevy, Anand Rajaraman, and Joann Ordille. Data integration: The teenage years. In *Proceedings of the 32nd international conference on Very large data bases*, pages 9--16, 2006.
- [74] F. T. Harold and K. Micki. *Information Security Management Handbook*, volume 2. CRC Press, 6th edition, 2015.
- [75] Mauricio A Hernández and Salvatore J Stolfo. The merge/purge problem for large databases. *ACM Sigmod Record*, 24(2):127--138, 1995.
- [76] Mauricio A Hernández and Salvatore J Stolfo. Real-world data is dirty: Data cleansing and the merge/purge problem. *Data mining and knowledge discovery*, 2(1):9--37, 1998.

- [77] Thomas N Herzog, Fritz J Scheuren, and William E Winkler. Data quality and record linkage techniques. Springer Science & Business Media, 2007.
- [78] HHS. Breach portal: Notice to the secretary of hhs breach of unsecured protected health information. https://ocrportal.hhs.gov/ocr/breach/breach_report.jsf.
- [79] HIPAA. Protected health information: What does phi include? <https://www.hipaa.com/hipaa-protected-health-information-what-does-phi-include>.
- [80] Rebecca Holman and Cees AW Glas. Modelling non-ignorable missing-data mechanisms with item response theory models. *British Journal of Mathematical and Statistical Psychology*, 58(1):1--17, 2005.
- [81] James Honaker, Gary King, Matthew Blackwell, et al. Amelia ii: A program for missing data. *Journal of statistical software*, 45(7):1--47, 2011.
- [82] Feng Honghai, Chen Guoshun, Yin Cheng, Yang Bingru, and Chen Yumei. A svm regression based approach to filling in missing values. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 581--587. Springer, 2005.
- [83] KC House Data. <https://www.kaggle.com/shivachandel/kc-house-data>, 1997. Accessed 7 March 2020.
- [84] George Hripcsak, Meryl Bloomrosen, Patti FlatleyBrennan, Christopher G Chute, Jim Cimino, Don E Detmer, Margo Edmunds, Peter J Embi, Melissa M Goldstein, William Ed Hammond, et al. Health data use, stewardship, and governance: ongoing gaps and challenges: a report from amia's 2012 health policy meeting. *Journal of the American Medical Informatics Association*, 21(2):204--211, 2014.
- [85] Jens Hußhn and Eyke Hußllermeier. Furia: an algorithm for unordered fuzzy rule induction. *Data Mining and Knowledge Discovery*, 19(3):293--319, 2009.
- [86] Caroline Humer and Jim Finkle. Your medical record is worth more to hackers than your credit card. *Reuters. com US Edition*, 24, 2014.

- [87] IBM and Ponemon Institute. 2015 cost of data breach study: Global analysis. Research report, IBM and Ponemon Institute, 2015.
- [88] Ali Inan, Murat Kantarcioglu, Elisa Bertino, and Monica Scannapieco. A hybrid approach to private record linkage. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 496--505. IEEE, 2008.
- [89] Ponemon Institute. Fifth annual benchmark study on privacy & security of health-care data. Research report, Ponemon Institute, 2015.
- [90] Anil Jadhav, Dhanya Pramod, and Krishnan Ramanathan. Comparison of performance of data imputation methods for numeric dataset. *Applied Artificial Intelligence*, 33(10):913--933, 2019.
- [91] Matthew A Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, 84(406):414--420, 1989.
- [92] Liang Jin, Chen Li, and S. Mehrotra. Efficient record linkage in large data sets. In *Proc. Eighth International Conference on Database Systems for Advanced Applications (DASFAA 2003)*, pages 137--146, March 2003.
- [93] J Howard Johnson. Rational equivalence relations. In *International Colloquium on Automata, Languages, and Programming*, pages 167--176. Springer, 1986.
- [94] David Kahn. *The Codebreakers: The comprehensive history of secret communication from ancient times to the internet*. Simon and Schuster, 1996.
- [95] Alexandros Karakasidis and Vassilios S Verykios. Reference table based k-anonymous private blocking. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pages 859--864, 2012.
- [96] Alexandros Karakasidis, Vassilios S Verykios, and Peter Christen. Fake injection strategies for private phonetic matching. In *Data Privacy Management and Autonomous Spontaneous Security*, pages 9--24. Springer, 2011.

- [97] Hillol Kargupta, Souptik Datta, Qi Wang, and Krishnamoorthy Sivakumar. On the privacy preserving properties of random data perturbation techniques. In Third IEEE international conference on data mining, pages 99--106. IEEE, 2003.
- [98] Rosemary Karmel, Phil Anderson, Diane Gibson, Ann Peut, Stephen Duckett, and Yvonne Wells. Empirical aspects of record linkage across multiple data sets using statistical linkage keys: the experience of the piac cohort study. *BMC health services research*, 10(1):41, 2010.
- [99] Diana Elizabeth Kendall, Rick Linden, and Jane Lothian Murray. *Sociology in our times*. Thomson Wadsworth Belmont, CA, 2005.
- [100] Abir Bin Ayub Khan, Mohammad Sheikh Ghazanfar, and Shahidul Islam Khan. Application of phonetic encoding for analyzing similarity of patient's data: Bangladesh perspective. In 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), pages 664--667. IEEE, 2017.
- [101] Shahidul Islam Khan, Md Mahmudul Hasan, Mohammad Imran Hossain, and Abu Sayed Md Latiful Hoque. namegist: a novel phonetic algorithm with bilingual support. *International Journal of Speech Technology*, 22(4):1135--1148, 2019.
- [102] Shahidul Islam Khan and Abu Sayed Latiful Hoque. Privacy and security problems of national health data warehouse: a convenient solution for developing countries. In *Networking Systems and Security (NSysS), 2016 International Conference on*, pages 1--6. IEEE, 2016.
- [103] Shahidul Islam Khan and Abu Sayed Md Latiful Hoque. Development of national health data warehouse bangladesh: Privacy issues and a practical solution. In 2015 18th International Conference on Computer and Information Technology (ICCIT), pages 373--378. IEEE, 2015.
- [104] Shahidul Islam Khan and Abu Sayed Md Latiful Hoque. An analysis of the problems for health data integration in bangladesh. In 2016 International Conference on Innovations in Science, Engineering and Technology (ICISSET), pages 1--4. IEEE, 2016.

- [105] Shahidul Islam Khan and Abu Sayed Md Latiful Hoque. Similarity analysis of patients' data: Bangladesh perspective. In 2016 International Conference on Medical Engineering, Health Informatics and Technology (MediTec), pages 1--5. IEEE, 2016.
- [106] Shahidul Islam Khan and Abu Sayed Md Latiful Hoque. Towards development of national health data warehouse for knowledge discovery. In Intelligent Systems Technologies and Applications, volume 385, 2 of Advances in Intelligent Systems and Computing, pages 413--421. Springer-Verlag, 2016.
- [107] Shahidul Islam Khan and Abu Sayed Md Latiful Hoque. Health data integration with secured record linkage: A practical solution for bangladesh and other developing countries. In 2017 International Conference on Networking, Systems and Security (NSysS), pages 156--161. IEEE, 2017.
- [108] Shahidul Islam Khan, Abu Sayed Md Latiful Hoque, and Mohammad Ullah. National health data warehouse bangladesh for remote health monitoring: Features, problems and privacy issues. In Remote Health Monitoring Workshop, 2016.
- [109] Shahidul Islam Khan and Abu Sayed Md Latiful Hoque. Digital health data: A comprehensive review of privacy and security risks and some recommendations. Computer Science Journal of Moldova, 24(2), 2016.
- [110] Abel N Kho, John P Cashy, Kathryn L Jackson, Adam R Pah, Satyender Goel, Joern Boehnke, John Eric Humphries, Scott Duke Kominers, Bala N Hota, Shannon A Sims, et al. Design and implementation of a privacy preserving electronic health record linkage tool in chicago. Journal of the American Medical Informatics Association, 22(5):1072--1080, 2015.
- [111] Hung-sik Kim and Dongwon Lee. Harra: fast iterative hashed record linkage for large-scale data collections. In Proceedings of the 13th International Conference on Extending Database Technology, pages 525--536, 2010.
- [112] Max Kuhn. A short introduction to the caret package. R Found Stat Comput, 1, 2015.

- [113] Hye-Chung Kum, Ashok Krishnamurthy, Ashwin Machanavajjhala, Michael K Reiter, and Stanley Ahalt. Privacy preserving interactive record linkage (ppirl). *Journal of the American Medical Informatics Association*, 21(2):212--220, 2014.
- [114] Mehmet Kuzu, Murat Kantarcioglu, Elizabeth Durham, and Bradley Malin. A constraint satisfaction cryptanalysis of bloom filters in private record linkage. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 226--245. Springer, 2011.
- [115] Sang Kyu Kwak and Jong Hae Kim. Statistical data preparation: management of missing values and outliers. *Korean journal of anesthesiology*, 70(4):407, 2017.
- [116] Pierre KY Lai, Siu-Ming Yiu, KP Chow, CF Chong, and Lucas Chi Kwong Hui. An efficient bloom filter based solution for multiparty private matching. In *Security and Management*, pages 286--292, 2006.
- [117] Rick L Lawrence and Andrea Wright. Rule-based classification systems using classification and regression tree (cart) analysis. *Photogrammetric engineering and remote sensing*, 67(10):1137--1142, 2001.
- [118] Choong Ho Lee and Hyung-Jin Yoon. Medical big data: promise and challenges. *Kidney research and clinical practice*, 36(1):3, 2017.
- [119] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 49--60, 2005.
- [120] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707--710, 1966.
- [121] M Paul Lewis. *Ethnologue: Languages of the world*. SIL international, 2018.
- [122] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 106--115. IEEE, 2007.

- [123] Yehida Lindell. Secure multiparty computation for privacy preserving data mining. In *Encyclopedia of Data Warehousing and Mining*, pages 1005--1009. IGI Global, 2005.
- [124] Ronan A Lyons, Kerina H Jones, Gareth John, Caroline J Brooks, Jean-Philippe Verplancke, David V Ford, Ginevra Brown, and Ken Leake. The sail databank: linking multiple health and social care datasets. *BMC medical informatics and decision making*, 9(1):3, 2009.
- [125] Paul Madley-Dowd, Rachael Hughes, Kate Tilling, and Jon Heron. The proportion of missing data should not be used to guide decisions on multiple imputation. *Journal of clinical epidemiology*, 110:63--73, 2019.
- [126] Claire Mathieu, Ocan Sankur, and Warren Schudy. Online correlation clustering. *arXiv preprint arXiv:1001.0920*, 2010.
- [127] Andrew McCallum, Kamal Nigam, and Lyle H Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 169--178, 2000.
- [128] Ralph C Merkle. One way hash functions and des. In *Conference on the Theory and Application of Cryptology*, pages 428--446. Springer, 1989.
- [129] S Micali, O Goldreich, and A Wigderson. How to play any mental game. In *Proceedings of the Nineteenth ACM Symp. on Theory of Computing, STOC*, pages 218--229, 1987.
- [130] Mohammed A Mohammed and Andrew Stevens. *The value of administrative databases*, 2007.
- [131] Noman Mohammed, Benjamin CM Fung, and Mourad Debbabi. Anonymity meets game theory: secure data integration with malicious participants. *The VLDB Journal*, 20(4):567--588, 2011.

- [132] Luis Carlos Molina, Lluís Belanche, and Àngela Nebot. Feature selection algorithms: A survey and experimental evaluation. In 2002 IEEE International Conference on Data Mining, 2002. Proceedings., pages 306--313. IEEE, 2002.
- [133] Arvind Narayanan and Vitaly Shmatikov. Fast dictionary attacks on passwords using time-space tradeoff. In Proceedings of the 12th ACM conference on Computer and communications security, pages 364--372, 2005.
- [134] Felix Naumann and Melanie Herschel. An introduction to duplicate detection. *Synthesis Lectures on Data Management*, 2(1):1--87, 2010.
- [135] Gonzalo Navarro. A guided tour to approximate string matching. *ACM computing surveys (CSUR)*, 33(1):31--88, 2001.
- [136] Jordi Nin, Victor Muntés-Mulero, Norbert Martínez-Bazan, and Josep-L Larriba-Pey. On the use of semantic blocking techniques for data cleansing and integration. In 11th International Database Engineering and Applications Symposium (IDEAS 2007), pages 190--198. IEEE, 2007.
- [137] U.S. Department of Health. Guidance regarding methods for de-identification of protected health information in accordance with the health insurance portability and accountability act (hipaa) privacy rule. <http://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/index.html>.
- [138] Bangladesh Bureau of Statistics. Population and housing census 2011 - volume 3: Urban area report. Research report, Bangladesh Bureau of Statistics, 2014.
- [139] Christine M O'Keefe, Ming Yung, Lifang Gu, and Rohan Baxter. Privacy-preserving data linkage protocols. In Proceedings of the 2004 ACM workshop on Privacy in the electronic society, pages 94--102, 2004.
- [140] Tomasz Orczyk and Piotr Porwik. Influence of missing data imputation method on the classification accuracy of the medical data. *Journal of Medical Informatics & Technologies*, 22, 2013.

- [141] Chaoyi Pang, Lifang Gu, David Hansen, and Anthony Maeder. Privacy-preserving fuzzy matching using a public reference table. In *Intelligent Patient Management*, pages 71--89. Springer, 2009.
- [142] Chaoyi Pang, David Hansen, et al. Improved record linkage for encrypted identifying data. *HIC 2006 and HINZ 2006: Proceedings*, page 164, 2006.
- [143] AJ Paverd, Andrew Martin, and Ian Brown. Modelling and automatically analysing privacy properties for honest-but-curious adversaries. *Tech. Rep.*, 2014.
- [144] Andrew Pavlo, Erik Paulson, Alexander Rasin, Daniel J Abadi, David J DeWitt, Samuel Madden, and Michael Stonebraker. A comparison of approaches to large-scale data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 165--178, 2009.
- [145] Kristiaan Pelckmans, Jos De Brabanter, Johan AK Suykens, and Bart De Moor. Handling missing values in support vector machine classifiers. *Neural Networks*, 18(5-6):684--692, 2005.
- [146] Olga Peled, Michael Fire, Rokach Lior, and Elovici Yuval. Matching entities across online social networks. *Neurocomputing*, 210:61--106, 2016.
- [147] Lawrence Philips. Hanging on the metaphone. *Computer Language*, 7(12 (December)), 1990.
- [148] Lawrence Philips. The double metaphone search algorithm. *C/C++ users journal*, 18(6):38--43, 2000.
- [149] Gregory Piatetsky-Shapiro, Ronald J Brachman, Tom Khabaza, Willi Kloesgen, and Evangelos Simoudis. An overview of issues in developing industrial data mining and knowledge discovery applications. In *KDD*, volume 96, pages 89--95, 1996.
- [150] Erhard Rahm and Hong Hai Do. Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, 23(4):3--13, 2000.

- [151] M Mostafizur Rahman and Darryl N Davis. Machine learning-based missing value imputation method for clinical datasets. In IAENG transactions on engineering technologies, pages 245--257. Springer, 2013.
- [152] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In Proceedings of the first instructional conference on machine learning, volume 242, pages 133--142. Piscataway, NJ, 2003.
- [153] Sean M Randall, Anna M Ferrante, James H Boyd, Jacqueline K Bauer, and James B Semmens. Privacy-preserving record linkage on large real world datasets. Journal of biomedical informatics, 50:205--212, 2014.
- [154] C. K. Reddy and C. C. Aggarwal. Healthcare data analysis. CRC Press, 2015.
- [155] MP Reddy, Bandreddi E Prasad, PG Reddy, and Amar Gupta. A methodology for integration of heterogeneous databases. IEEE transactions on knowledge and data engineering, 6(6):920--933, 1994.
- [156] Soundex System - National Archives, 2007. Accessed 4 October 2018.
- [157] jellyfish - a python library for doing approximate and phonetic matching of strings. <https://github.com/jamesturk/jellyfish>, 2018. Accessed 4 October 2018.
- [158] Open source name database. <https://github.com/smashew/NameDatabases>, 2013. Accessed 4 October 2018.
- [159] Unicode Bengali name collection. <https://bit.ly/2FZEmZV>, 2017. Accessed 4 October 2018.
- [160] World population prospects - United Nations. <https://population.un.org/wpp/DataQuery/>, 2017. Accessed 4 October 2018.
- [161] Bengali (Bangla) - University of Washington. <https://asian.washington.edu/fields/bengali-bangla>, 2017. Accessed 4 October 2018.
- [162] International Mother Language Day - UNESCO, 2017. Accessed 4 October 2018.

- [163] Beyond the Top 1000 Names - USA Social Security Administrations. <https://www.ssa.gov/oact/babynames/limits.html>, 2017. Accessed 4 October 2018.
- [164] Frequently Occurring Surnames from the Census 2000 - US Census Bureau. https://www.census.gov/topics/population/genealogy/data/2000_surnames.html, 2014. Accessed 4 October 2018.
- [165] Match rating approach - Wikipedia, 2017. Accessed 4 October 2018.
- [166] Bernard Peter Robichau. Healthcare Information Privacy and Security: Regulatory Compliance and Data Security in the Age of Electronic Health Records. Apress, 1st edition, 2014.
- [167] Steve Robinson and John W Polak. Modeling urban link travel time with inductive loop detector data by using the k-nn method. *Transportation research record*, 1935(1):47--56, 2005.
- [168] Peter Schmitt, Jonas Mandel, and Mickael Guedj. A comparison of six methods for missing data imputation. *Journal of Biometrics & Biostatistics*, 6(1):1, 2015.
- [169] Rainer Schnell, Tobias Bachteler, and Stefan Bender. A toolbox for record linkage. *Austrian Journal of Statistics*, 33(1&2):125--133, 2004.
- [170] Rainer Schnell, Tobias Bachteler, and Jörg Reiher. Privacy-preserving record linkage using bloom filters. *BMC medical informatics and decision making*, 9(1):41, 2009.
- [171] Rainer Schnell and Christian Borgs. Building a national perinatal data base without the use of unique personal identifiers. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 232--239. IEEE, 2015.
- [172] Krebson Security. Premera blue cross breach exposes financial, medical records. <http://krebsonsecurity.com/2015/03/premera-blue-cross-breach-exposes-financial-medical-records>.
- [173] R. D Snee. Hair and eye color of statistics students. <https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/HairEyeColor.html>.

- [174] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557--570, 2002.
- [175] Gregory Tauer, Ketan Date, Rakesh Nagi, and Moises Sudit. An incremental graph-partitioning algorithm for entity resolution. *Information Fusion*, 46:171--183, 2019.
- [176] Huynh Cao Tri. Unesdoc report: Principal characteristics of the least developed countries. <http://unesdoc.unesco.org/images/0004/000483/048315Eb.pdf>.
- [177] Isaac Triguero, Sergio González, Jose M Moyano, Salvador García López, Jesús Alcalá Fernández, Julián Luengo Martínez, Alberto Fernández Hilario, Jesús Díaz, Luciano Sánchez, Francisco Herrera Triguero, et al. Keel 3.0: an open source software for multi-stage analysis in data mining. 2017.
- [178] Chun-Wei Tsai, Chin-Feng Lai, Han-Chieh Chao, and Athanasios V Vasilakos. Big data analytics: a survey. *Journal of Big data*, 2(1):21, 2015.
- [179] Car Evaluation Data Set. <http://archive.ics.uci.edu/ml/datasets/Car+Evaluation>, 1997. Accessed 4 February 2020.
- [180] Esko Ukkonen. Approximate string-matching with q-grams and maximal matches. *Theoretical computer science*, 92(1):191--211, 1992.
- [181] Naushad UzZaman and Mumit Khan. A bangla phonetic encoding for better spelling suggestions. Technical report, BRAC University, 2004.
- [182] Naushad UzZaman and Mumit Khan. A double metaphone encoding for bangla and its application in spelling checker. Technical report, BRAC University, 2005.
- [183] Dinusha Vatsalan, Peter Christen, and Vassilios S Verykios. A taxonomy of privacy-preserving record linkage techniques. *Information Systems*, 38(6):946--969, 2013.
- [184] Dinusha Vatsalan et al. Scalable and approximate privacy-preserving record linkage. 2014.

- [185] Dinusha Vatsalan, Ziad Sehili, Peter Christen, and Erhard Rahm. Privacy-preserving record linkage for big data: Current approaches and research challenges. In *Handbook of Big Data Technologies*, pages 851--895. Springer, 2017.
- [186] Verizon. Protected health information data breach report. Research report, Verizon, 2015.
- [187] Vassilios S Verykios, Ahmed K Elmagarmid, and Elias N Houstis. Automating the approximate record-matching process. *Information sciences*, 126(1-4):83--98, 2000.
- [188] Vassilios S Verykios, Alexandros Karakasidis, and Vassilios K Mitrogiannis. Privacy preserving record linkage approaches. *International Journal of Data Mining, Modelling and Management*, 1(2):206--221, 2009.
- [189] Gerko Vink, Laurence E Frank, Jeroen Pannekoek, and Stef Van Buuren. Predictive mean matching imputation of semicontinuous variables. *Statistica Neerlandica*, 68(1):61--90, 2014.
- [190] Susan C Weber, Henry Lowe, Amar Das, and Todd Ferris. A simple heuristic for blindfolded record linkage. *Journal of the American Medical Informatics Association*, 19(e1):e157--e161, 2012.
- [191] Jens H Weber-Jahnke and Christina Obry. Protecting privacy during peer-to-peer exchange of medical documents. *Information systems frontiers*, 14(1):87--104, 2012.
- [192] Steven Euijong Whang and Hector Garcia-Molina. Entity resolution with evolving rules. *Proceedings of the VLDB Endowment*, 3(1-2):1326--1337, 2010.
- [193] Steven Euijong Whang and Hector Garcia-Molina. Incremental entity resolution on rules and data. *The VLDB Journal—The International Journal on Very Large Data Bases*, 23(1):77--102, 2014.
- [194] WHO. Global health observatory data repository: Life expectancy- data by country. Technical report, World Health Organization, type =.

- [195] William E Winkler. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. 1990.
- [196] William E Winkler et al. Improved decision rules in the fellegi-sunter model of record linkage, volume 56. Citeseer, 1993.
- [197] Raymond E Wright. Logistic regression. 1995.
- [198] Mohamed Yakout, Mikhail J Atallah, and Ahmed Elmagarmid. Efficient private record linkage. In 2009 IEEE 25th International Conference on Data Engineering, pages 1283--1286. IEEE, 2009.
- [199] Ji-Jiang Yang, Jian-Qiang Li, and Yu Niu. A hybrid solution for privacy preserving medical data sharing in the cloud environment. *Future Generation computer systems*, 43:74--86, 2015.
- [200] Yiming Yang. An evaluation of statistical approaches to text categorization. *Information retrieval*, 1(1-2):69--90, 1999.
- [201] Andrew Chi-Chih Yao. How to generate and exchange secrets. In 27th Annual Symposium on Foundations of Computer Science (sfcs 1986), pages 162--167. IEEE, 1986.
- [202] Shichao Zhang, Chengqi Zhang, and Qiang Yang. Data preparation for data mining. *Applied artificial intelligence*, 17(5-6):375--381, 2003.
- [203] Y. Zhang and C. Poon. Editorial note on bio, medical and health informatics. *IEEE Transactions on Information Technology in Biomedicine*, 14(3):543--545, 2010.
- [204] Zhongheng Zhang. Missing values in big data research: some basic skills. *Annals of translational medicine*, 3(21), 2015.
- [205] Zheng Zhao and Huan Liu. Searching for interacting features in subset selection. *Intelligent Data Analysis*, 13(2):207--228, 2009.