Privacy Preserving Software Engineering for Data Driven Development

by

Karan Naresh Tongay
B.E., Savitribai Phule Pune University, 2017
A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

Privacy Preserving Software Engineering for Data Driven Development

by

Karan Naresh Tongay
B.E., Savitribai Phule Pune University, 2017

Supervisory Committee

Dr. Neil Ernst , Supervisor
(Department of Computer Science)

Dr. Sean Chester , Departmental Member
(Department of Computer Science)

Supervisory Committee

Dr. Neil Ernst , Supervisor
(Department of Computer Science)

Dr. Sean Chester , Departmental Member
(Department of Computer Science)

ABSTRACT

The exponential rise in the generation of data has introduced many new areas of research including data science, data engineering, machine learning, artificial intelligence to name a few. It has become important for any industry or organization to precisely understand and analyze the data in order to extract value out of the data. The value of the data can only be realized when it is put into practice in the real world and the most common approach to do this in the technology industry is through software engineering. This brings into picture the area of privacy oriented software engineering and thus there is a rise of data protection regulation acts such as GDPR (General Data Protection Regulation), PDPA (Personal Data Protection Act), etc. Many organizations, governments and companies who have accumulated huge amounts of data over time may conveniently use the data for increasing business value but at the same time the privacy aspects associated with the sensitivity of data especially in terms of personal information of the people can easily be circumvented while designing a software engineering model for these types of applications. Even before the software engineering phase for any data processing application, often times there can be one or many data sharing agreements or privacy policies in place. Every organization may have their own way of maintaining data privacy practices for data driven development. There is a need to generalize or categorize their approaches into tactics which could be referred by other practitioners who are trying to integrate data privacy practices into their development. This qualitative study provides an understanding of various approaches and tactics that are being practised within the industry for privacy preserving data science in software engineering, and discusses a

tool for data usage monitoring to identify unethical data access. Finally, we studied strategies for secure data publishing and conducted experiments using sample data to demonstrate how these techniques can be helpful for securing private data before publishing.

Contents

Supervisory Committee **ii**

Abstract **iii**

Table of Contents **v**

List of Tables **viii**

List of Figures **ix**

Acknowledgements **xi**

Dedication **xii**

1 Introduction **1**

1.1 Motivation 1
1.2 Research questions 2
1.3 Data usage monitoring tool 3
1.4 Industrial survey 3
1.5 Techniques behind private data publishing 4
1.6 Contributions 5
1.7 Thesis overview 6

2 Case study introduction and background **7**

3 Data usage monitoring tool **10**

3.1 Introduction 10
3.2 Related work 12
3.2.1 Data Collection 12
3.2.2 Log Preprocessing 13

3.2.3 Rule Based Control and User Pattern Analysis 15
3.2.4 Machine Learning Approach 16
3.2.5 Summary 18
3.3 Our implementation 19
3.3.1 Design Challenges 20
3.3.2 System architecture 21
3.3.3 Postgres log structure 23
3.3.4 Rule encoding phase 23
3.3.5 Machine learning model training 25
3.3.6 Evaluation 26
3.4 Chapter summary 28

4 Industrial survey **29**
4.1 Introduction 29
4.2 Related work 30
4.2.1 Introduction 30
4.2.2 Developer and user viewpoint on data privacy 30
4.2.3 Privacy education 31
4.2.4 Organizational climate 32
4.2.5 Challenges to embed data privacy in the software 33
4.2.6 Gap we address between the literature and our study 35
4.3 Survey on data privacy as a quality attribute in the industry 36
4.3.1 Methodology and Demographics 36
4.3.2 Categorization of participant responses 39
4.3.3 Validity Threats 44
4.4 Chapter summary 44

5 Secure data publishing techniques **46**
5.1 Introduction 46
5.2 ℓ -diversity and T-closeness 48
5.2.1 Introduction 48
5.2.2 Objective 48
5.2.3 Approach for ℓ-diversity 49
5.2.4 Approach for t-closeness 50
5.3 Differential Privacy 53

5.3.1 Introduction 53
5.3.2 Mechanism of differential privacy 55
5.3.3 Mathematical foundation 57
5.3.4 Privacy budget composition 58
5.3.5 Our implementation 59
5.3.6 System architecture and specifications 60
5.4 Limitations 65
5.5 Chapter summary 65

6 Discussion **67**
6.1 Audits and access control is the preferred tactic to monitor data access within organizations 69
6.2 Understanding and implementing the data privacy regulations at work is still a challenge 70
6.3 Anonymizing data and control over data sharing helps in promoting secure data sharing environment 71
6.4 Findings 72

7 Conclusion **75**

Bibliography **78**

List of Tables

Table 3.1 Postgres DB Log Format 24
Table 3.2 ML Dataset 25
Table 3.3 ML Dataset 26

Table 4.1 Participant Role and Experience 37

Table 5.1 Example dataset where 'Answer' is the sensitive attribute 55
Table 5.2 Disabling privacy budget when average of results is close to the original answer by +/- 0.01 64

List of Figures

Figure 3.1 Agreement between data provider and trusted third party . . 10
Figure 3.2 Possibility of data re-identification and its unethical monetization 11
Figure 3.3 Proposed method 19
Figure 3.4 System architecture 21
Figure 3.5 Unstructured Postgres logs 23
Figure 3.6 Detected anomalies 27

Figure 4.1 Use of programming languages 38
Figure 4.2 Do you use user's data for ML training? 38
Figure 4.3 Use of third party tools 39
Figure 4.4 Adoption of privacy regulations at work 40
Figure 4.5 Survey response counts per categorization 41

Figure 5.1 System architecture extension - upper left 47
Figure 5.2 ℓ-diversity python code 50
Figure 5.3 Data loss for each threshold value 51
Figure 5.4 Equivalence class 2 52
Figure 5.5 Equivalence class 3 53
Figure 5.6 T-closeness of sensitive values within equivalence classes . . . 54
Figure 5.7 Spinners to represent the overview concept of differential privacy mechanism 56
Figure 5.8 Output of each spinner after 100 spins 57
Figure 5.9 Laplace noise addition using Numpy 58
Figure 5.10 Differential privacy prototype 60
Figure 5.11 Admin epsilon selection and query results after 10 queries . . 61
Figure 5.12 Admin epsilon selection and query results after 100 queries . . 62
Figure 5.13 Admin epsilon selection as 0.1 (highest noise) and query results after 10 queries 62

Figure 5.14 Count query execution by the user, privacy budget not yet exhausted 63
Figure 5.15 The privacy budget exhausted by the user and is now disabled to query on that asset 64

Figure 6.1 System architecture - replicated from section 3.4 67

ACKNOWLEDGEMENTS

First and foremost I would like to express my sincere gratitude to my graduate advisor Dr. Neil Ernst for his continuous support to my M.Sc. study and related research. I would like to thank him for his patience, motivation and immense knowledge. His guidance helped me in all the time of research and writing this thesis. Its hard to imagine having better advisor and mentor for my M.Sc study. Its a dream come true for me and I thank him from the bottom of my heart.

Besides my advisor, I would like to thank Dr. Sean Chester for being on the committee and being my supervisor for directed studies in Privacy Preserving Data Science coursework. I thank Dr. Jens Weber for giving me an opportunity to collaborate with his research lab during the engage 18 project. I am grateful to Zane for being a mentor, providing insightful comments and constant encouragement.

I would also like to thank Malatest and Shift - Redbrick for providing me an opportunity to work as a research partner and ICBC for the co-op opportunity, where I got practical experience in the field of my thesis.

Moreover, I thank my fellow lab members for making me feel home and motivated at workplace and for all the activities and fun we have had since the last two years. Special thanks to Dr. Hausi Muller and Dr. Ulrike Stege for providing the space in Rigi Lab during my M.Sc and involving me in all the fun and professional activities within the lab. I also want to extend my special thanks to Dr. Bill Bird for his trust in me during the work-study project and other TA responsibilities.

All of this wouldn't have been possible without the equal efforts from the administrative staff of the University of Victoria and the Department of Computer Science. I thank them for all the administrative services they provided me during my M.Sc.

Also, I take this opportunity to thank my friends outside research lab Prakriti Sharma, Abhishek Kumar Bojja, Souvik Maitra, Vikas Prasad, Adeshina Alani, Yugansh Gupta and Shirley Wang for being a part of my great journey. I am grateful to all of you for filling my life outside the lab and for being a constant pulse of motivation.

Moreover, I would like to thank my brother Ninad Tongay and my mother Nirmala Tongay for supporting me morally and spiritually throughout writing this thesis and my life in general.

Karan Tongay

DEDICATION

This thesis work is dedicated to my late father Dr. Naresh Tongay, mother Nirmala Tongay, brother Ninad Tongay, all my well wishers, family, friends and my mentors.

Chapter 1

Introduction

1.1 Motivation

The rise of data and its prime importance in helping with making data driven decisions have given a new direction to the field of software engineering. Data is rightly defined as the new "oil" [5]. It helps any organization make efficient and reliable decisions for themselves and their users which is adding tremendous value in modern life. From search engines to online shopping websites, the way we used to interact with these services has drastically changed over the decade, all thanks to the humongous amount of data being generated every second all around the world. This also means that the large amount of generated data is being collected, stored and processed by the service providing organizations [5]. Moreover, the practice of data collection has been less transparent to the users. Due to this, the advent of data protection regulation acts like GDPR (General Data Protection Regulation) act, PDPA (Personal Data Protection Act) etc. was eminent. Even though these acts are in place and also the privacy policies of respective organizations where they mention their data collection practices, it is another challenge in software engineering to make the software algorithm follow the context of the privacy policies. From the GDPR context, one approach would be to monitor data usage to keep a track of how the personal data is being accessed. We built a data usage monitoring tool to contribute in this direction. Additionally, we decided to know what are the tactics followed in industry to address data privacy challenges and also to know the applicability of our data usage monitoring tool in the industrial context. Furthermore, in early 2020, the United States Census Bureau implemented a new gold standard in data privacy protection called differential privacy

and the 2020 census data was protected using differential privacy when it was released. This being a motivation, we realized that along with data usage monitoring, having some control over data sharing at a initial level would help in addressing data sharing concerns within data driven organizations which led us to extend our existing data usage monitoring tool with an additional layer of control over secure data publishing. Altogether, we were able to build an end to end tool from supporting secure data publishing mechanism using techniques like ℓ-diversity, t-closeness and differential privacy to monitoring the usage of data through database logs. We will be discussing about our tool in detail in the subsequent chapters.

1.2 Research questions

The research work started by exploring the solution to the problem of monitoring fair usage of data by the data consumers. One of the best way to achieve this was through monitoring database logs [21]. The data usage could be restricted using access control mechanisms of a database, and idea of log monitoring is similar to conducting data access audits. We decided to name this tactic as 'audits and access control'. Similarly, after developing the prototype of our tool, we tried to explore additional tactics which data driven practitioners use in the industry sector by conducting an industrial survey. Furthermore, we realized the role of our data usage monitoring tool in the industrial paradigm through this survey and decided to extend our tool to support secure data publishing making it an end-to-end secure data publishing and data usage monitoring tool. This study discusses and provides meaningful contributions for both researchers and practitioners by answering the research questions below:

Q 1. How machine learning can be used as audits and access control tactic to maintain the quality of data privacy?
Q 2. Which tactics do the data driven practitioners in the industry follow or suggest to ensure data privacy?
Q 3. Which techniques could be used to secure sensitive information before data publishing to gain better control over data sharing?

1.3 Data usage monitoring tool

Our research in this direction started by working on a problem statement to develop a data usage monitoring tool. After conducting several meetings, we defined the initial problem which was to monitor fair usage of outsourced data to the data consumers, especially in the context of health data sharing. In these initial set of requirements, the data was not supposed to be anonymized and we had to assume that the data will be shared in the raw format and accessing this data should be monitored. We began by understanding the challenges associated with data sharing in the healthcare industry and how important it is to monitor the usage of shared data in order to detect or prevent misuse of individual's personal information. Further, in order to simulate the real-life setting, we derived an initial set of rules for the data sharing agreement, synthetic data set to be shared and queried our postgres database multiple times adhering to the specified rules in data sharing agreement to generate sufficient amount of logs. After this initial setup, we started designing our architecture to address the specified problem of monitoring fair data usage. After understanding the problem as a whole, we came to a realization that it is an anomaly detection problem. The anomalies were the database logs that violated the data license agreement. While designing our architecture, we started to build a solution/tool to address the problem. The tool is supposed to identify unethical access to the PII (Personally Identifiable Information) that violates the data license agreement by monitoring postgres database logs. This led us to build a platform that tries to address the data sharing and PII access control challenges which we discuss in Chapter 2.

1.4 Industrial survey

After building the data usage monitoring tool for the requirements specified by our research partner, we decided to learn how data privacy is addressed in different industry sectors which generated curiosity within us to understand the data privacy challenges by conducting interview style survey with data driven developers. Objective was to understand the tactics they follow to maintain data privacy and to find out if our data usage monitoring tool has any applicability in the industry. We conducted a literature review in order to study the existing survey and interview based research on industrial practices and awareness among developers with respect to addressing data privacy challenges within their organization. We further identified an important gap between

our research goals and the current literature; all of the studies which we reviewed targetted group of developers as a whole, but they may or may not have involved data driven developers in the study. We believed data driven developers specifically would be the right audience for this kind of study and conducting a interview-style survey within the industry among this community would allow us to gain fair amount of information on data privacy practices within the industry. Therefore, we decided to engineer a survey that targeted data driven developers and conducted it across the decision makers and practitioners within the industry. We began by designing questions that aligned with the goals of our research. The survey had questions which were focused on understanding the demographics of an individual, the data privacy practices at work and the challenges they face with respect to maintaining and achieving data privacy at work. The survey consisted of both multiple choice and open ended questions. After all the survey responses were received, the open ended responses were codified into several categories and multiple choice questions were used to draw direct insights through visualisations (charts). After coding all the open ended responses we were able to extract similar codings together and further categorised them. Out of the many responses we analysed, it was insightful to observe that majority of the participants indicated access control frequently. This made us realize that the tool which we developed for data usage monitoring has applicability in the industries that involve monitoring and access control over PII.

1.5 Techniques behind private data publishing

The idea of data usage monitoring based on the requirements was realized into a working prototype. Although, we decided to extend our architecture further. Sharing personal data and monitoring access could be useful, but anonymizing or aggregating the data in first place before sharing the data would bring control to the entire data sharing and monitoring system. The motivation was ignited after the US government decided to protect the 2020 census results with the help of differential privacy, which they defined as a new gold standard in data privacy protection. Another motivation was insights we gained from our survey, out of which one of them indicated an interest towards quantifying data privacy through a measurement criteria for bringing control in data sharing. We did experiments using different techniques for privacy preserving data publishing. Eventually, we realized that these techniques can be used to define a quantifying measure for data privacy. This component later became an extension

module to our data usage monitoring tool architecture.

1.6 Contributions

Contribution 1 - Proposed system architecture for data usage monitoring tool

After studying several methods for monitoring data access control using machine learning, we learnt that log monitoring and anomaly detection is the reliable way to achieve it. We studied methods of data collection, log processing, rule based control and use of machine learning for log monitoring. After gaining an understanding of the foundation, we realized identified the potential of machine learning to solve the problem of data usage monitoring for our use case. We decided to build a simple yet generic tool which would address the issue of monitoring data access of the data consumers which may include a formal data sharing agreement between the parties. In this way, through our system architecture, we attempted to demonstrate how machine learning can be helpful in maintaining quality of data privacy through data access monitoring and serve as a key component for "Audits and access control" tactic to maintain vigilance over data sharing among data driven developers.

Contribution 2 - Identifying data privacy tactics within the industry

After developing the data usage monitoring tool for our use case described in chapter 2, we decided to find out the applicability of our tool in the broader industry sector through a survey. The industrial survey was targeted towards key decision makers and data driven developers in within the industry. Using the design science approach, we found out that our tool has a scope of application within the industry. Majority of the responses of our participants were in the category of audits and access control followed by data related operations. After thematic analysis and categorization of responses, we found 4 key tactics practiced by the data driven developers in the industry, which are 'Audits and access control', 'Data related operations', 'Privacy awareness' and 'Machine learning protocol'. Although we acknowledge the low number of participants in our study, majority of the participants were decision makers in their organization which added quality into the findings of our study. Based on the analysis of their responses, we learnt that 'Audits and access control' was the most discussed tactic among the industrial participants thereby indicating a scope of applicability of our data access monitoring tool to address this problem.

Contribution 3 - Extending our system architecture by practically implementing theoretical data privacy techniques
We studied 3 data privacy techniques during the course of this research. Two of them (ℓ-diversity and t-closeness) being data anonymization methods and the third (differential privacy) being the controlled data publishing mechanism. The motivation to study these techniques came from our need to extend our existing data usage monitoring tool with secure data publishing mechanism to implement an end-to-end application from secure data publishing to monitoring of data usage. We further understood how these techniques could be used together along with our existing data usage monitoring tool and make the end-to-end data publishing and usage monitoring process more secure and controlled.

1.7 Thesis overview

Let's go through what you can expect in each of the chapters in this thesis:
Chapter 2 introduces the case study and prerequisites we realized as a foundation for our data usage monitoring tool, data anonymization and differential privacy implementation. We introduce Synthea [3] the synthetic dataset which we used as a reference throughout our thesis and it served as a sample development database for our data usage monitoring tool, data anonymization implementation and differential privacy.
Chapter 3 focuses on background and concepts that are required to understand the underlying context of the problem and presents our implementation of data license monitoring tool as a tactic for data privacy quality attribute.
Chapter 4 describes our findings from the industrial survey that we conducted among the key decision makers within their respective organization to know about data privacy approaches, tactics and challenges.
Chapter 5 describes and demonstrates secure data sharing approaches and techniques to help reduce privacy concerns for data publishing.
Chapter 6 starts by discussing practical and research implications of this research and ends by summarizing the thesis and identifying the limitations.
Chapter 7 presents our conclusion of this research study and highlights the future work.

Chapter 2

Case study introduction and background

Our research partner reached out to us with the problem of data usage monitoring of shared health data. The health data will be shared to a third party using a data sharing agreement and the data owner would ideally be a health authority. The research partner already had a cloud data sharing architecture in place, although research was needed in the direction of monitoring the shared health data. In this thesis, we attempted to demonstrate end-to-end privacy preserving data science methodology i.e. from determining privacy of the data in quantifiable terms for publishing the data to monitoring the data usage. Our main focus in this research was studying the tactics that help with the data privacy quality attribute.

For the purpose of this research, we used one common data set to demonstrate our experiments. We generated synthetic patient health data using Synthea [3] - an open source synthetic patient generator that models the medical history of synthetic patients and we have considered Synthea [3] as our data provider entity for this experiment. Generally, data provider can be defined as a framework for making the data available to the data requester from the source.

The Synthea dataset served as our input data source. Based on the requirements received, anonymizing the data was not a requirement, rather we were supposed to be careful not to assume any kind of data anonymization taking place before data sharing. Therefore, we did not anonymize or quantify the privacy of our data initially for our developing data usage monitoring tool. There was an assumption of having a sample data license agreement which prescribed certain rules of accessing the data.

Any query violating the below rules should be flagged as violation of data license agreement. For the purpose of research, we had three simple rules in our data sharing agreement:

- Only these users are allowed to use the synthea database: [‘karan’, ‘postgres’].
- The users should not access the “patients” table.
- The users should not access the patient information using foreign key from allergies, immunizations, observations, encounters or procedures table.

For example:

Valid query would be:

```
select code, description from allergies;
```

Violating query would look like:

```
select patient.first_name, description from allergies inner join
patients on allergies.patient = patients.id;
```

In order to automate this process, we trained a machine learning model using one-class SVM as we looked at it as a anomaly detection problem. Anomaly detection is an approach of identifying rare or novel events within the data which raises suspicions by significantly differing from the majority of the data [28]. We describe the specifications of our model further in Chapter 2. Finally, we extended this study by studying techniques to help reduce privacy concerns before sharing the data. Studying them also helped us to define a measurable quantity to define the privacy level of the data. We designed experiments using the same Synthea dataset for demonstrating secure data publishing techniques for data privacy. We describe them in Chapter 5:

- L-diversity and T-closeness
- Differential privacy

Additionally, we studied the developer and user point of view on data privacy. There is a need for more research into data privacy in software engineering to reduce the responsibility on users to understand how the software works and handles their information [18]. Furthermore, organizational climate promotes behaviour that is inconsistent with the defined privacy policies or regulations [17]. Although, we realized

that majority of these existing studies were focused on general audience of software developers and not particularly on data driven developers who frequently deal with the data. We decided to address this gap and conduct a survey to understand data privacy tactics of data driven developers within the industry which we discuss in chapter 4. Every organization or a project team may have their own set of tactics to address data privacy in software engineering. Although, there is no formal existing tactics tree for data privacy in software engineering, but the tactics related to data oriented strategies, such as data minimization or data anonymization and process oriented strategies such as data usage audits, access control or privacy awareness among developers may help in ensuring data privacy in software engineering. We attempted to group the responses of the participants of our survey and tried to come up with four key tactics used within the industry which we explain in chapter 4.

Chapter 3

Data usage monitoring tool

3.1 Introduction

Monitoring the fair usage of the outsourced data is a challenge for most of the data providers. Data has lately become a precious gem for many huge organizations and governments. It gives them insight for understanding market trends, consumer choices and sentiments which helps them drive their work in a right direction. Not all business entities or governments have a well established team or department for carrying out data analysis and they might end up outsourcing their data to trusted third parties to generate insights out of their data. Sometimes, the data providers can be hospitals, pharmacy, clinics or similar entities who own the private data of the citizens. Maintaining the confidentiality of such data is a primary responsibility of these entities and therefore they must anonymize the data before it is leased out. Figure 3.1 depicts that they lease their data to the trusted third parties with the good intentions of extracting valuable insights out of their data so that they could develop their services to improve the healthcare of their patients.

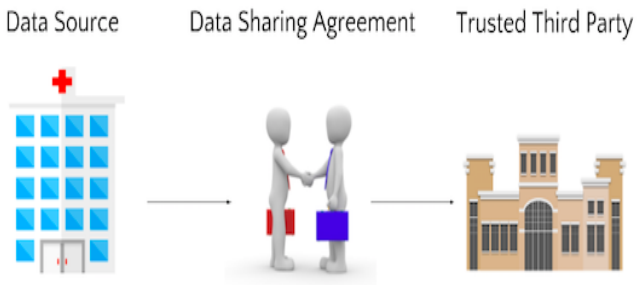

Figure 3.1: Agreement between data provider and trusted third party

In the process of outsourcing the data, a data license agreement can be signed between the data source and the trusted third party so as to limit the amount of data to share because the data providers may not have total confidence in what the third-party will be doing with the data. The data license prescribes the boundaries and fair usage policy for the data. Though there is a formal agreement, it is still a challenge to monitor activities of the licensee and be assured that the data is being used fairly. The license violations can be performed by an outsider or an insider [28]. Personal information leakage can sometimes be unintentional and may be caused due to an accident [28].

Few data providers choose to monitor it manually through human resources but it can be tedious as well as expensive. Therefore, there is a need for an automated method which can be trusted upon and help the licensors identify the violations within time in order to prevent serious compromise of the personal information.
Another important concern associated with sharing data is control over data re-identification. If the trusted third party identifies some personal interest in the data, it is a possibility that they collect private data from different data sources viz. hospitals, pharmacy, clinics etc. in order to re-identify the individual from different anonymous data. As a consequence, this data could be used for targeted business, digital marketing and several other business activities which compromises the confidentiality of the private data of the patients. Figure 3.2 represents the possible risk after sharing data with the trusted third parties which can be accidental or intentional.

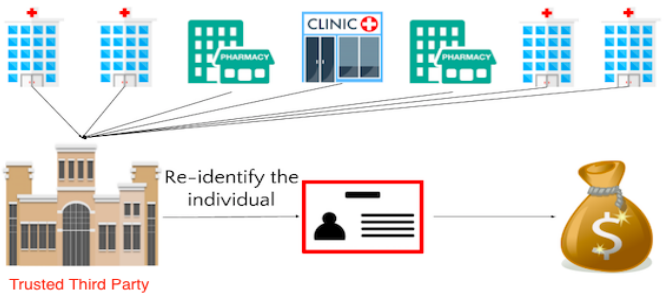

Figure 3.2: Possibility of data re-identification and its unethical monetization

Such compromise of confidentiality can be termed as a violation of the Data License Agreement. One of the best source, to identify whether a violation has taken place or can happen in future, is database system logs. These violations are often termed as anomalies which can be observed in the database access patterns. Log data

is a valuable and important object for understanding system status and performance issues; therefore, the several system logs are naturally excellent source of information for anomaly detection and online log monitoring [11]. Logs contain inevitable information and track record of every single activity in the database chronologically. In this literature review, I discuss different methods to process and analyze logs along with machine learning approaches to identify anomalies in the database access logs and how these methods can make a significant impact in identifying direct or indirect data license violations. I also discuss, how it can be used to aid the data sharing concerns associated with private data. After comparing and contrasting different methodologies I am combining the best features of each method and coming up with a single approach which best addresses this problem.

3.2 Related work

3.2.1 Data Collection

Collecting appropriate data for performing log analysis for research can be very challenging. Using real, non-anonymized data raises a variety of legal, ethical, and business issues and therefore sometimes we need to turn towards proxy data sets and synthetic data. Despite a widespread use of synthetic data to test classification systems, producing synthetic data that achieves a high level of human realism is a much more difficult problem [15]. This is because even if we create synthetic data, it still might miss several very important dimensions. A single piece of data that may be valid on its own may be inconsistent in relation to other pieces of data [15]. Glasser and Lindauer [15] introduce a methodology towards generation of synthetic yet realistic data. The research community found their generated data to demonstrate many important characteristics of realism. Even though fully synthetic data can't replace real data along with other benefits, it can significantly lower the barriers to entry into research requiring such data and provide the type of experimental control necessary to help establish a solid scientific foundation for such research [15]. For the purpose of this research we are using Synthea [3] which is an open source tool for synthetic patient data generation.

3.2.2 Log Preprocessing

Log preprocessing is the one of the primary challenging step in log analysis. Logs collect large amounts of relevant information about what is happening in a system, at least if the underlying systems and applications are properly configured to do so [24]. These logs are the potential source of detecting anomalies which in our context is defined as data license violation. Database log analysis plays a significant role in anomaly detection and log messages recording detailed system runtime information has become an important data analysis object accordingly [8]. The log data preparation process is the most time consuming and intensive step [27]. The volume of data generated in the logs can be really large [12]. We should parse unstructured or semi-structured logs into structured data and extract features before log analysis [37]. The rest of this section is the discussion on several existing log preprocessing methods proposed in the research community.

The state-of-the-art log parsing method is represented by Spell which is an unsupervised streaming parser that parses incoming log entries in an online fashion [11]. DeepLog uses log keys and also metric values in a log entry for anomaly detection and it is able to capture different types of anomalies [11]. Their past work on log analysis has discarded timestamp and/or parameter values in a log entry, and only used log keys to detect anomalies. Each log key is the execution of a log printing statement in the source code. They propose to model anomaly detection in a log key sequence as a multiclass classification problem, where each distinct log key defines a class [11]. The intuition is that log keys in the same task always appear together, but log keys from different tasks may not always appear together as the ordering of tasks is not fixed during multiple executions of different tasks. This allows us to cluster log keys based on co-occurrence patterns, and separate keys into different tasks when co-occurrence rate is low [11]. Marchi et al. propose the concept of encoding the input data using their autoencoder. Later, the reconstruction error is calculated between the input and the output of the autoencoder is used to detect novel events and these novel events can be anomalies [23].

Du and Cao introduce another method of log preprocessing. According to them, there is a big difference between clustering log messages and ordinary data variables, for log messages do not have a concept of dimensionality, while ordinary data is a vector consisting of several features [12]. Their first step of log preprocessing in their two step anomaly detection involves a categorization method that categorizes

log data into behavior sequences in an appropriate granularity, via a hierarchical clustering algorithm which makes use of features extracted from log messages [12]. In the second step, they generate behavior pattern sets from clustered messages and assign an anomaly score to new log sequences according to the relation between the log sequences series of behavioral features extracted from log messages in a periodic time interval. To categorize message records into clusters, we raised a hierarchical clustering algorithm that makes use of the log payload and other fields such as the log level [12].

Feature engineering

Lopez and Sartipi introduce the method of feature engineering from the logs. The process of feature engineering may involve mathematical transformation of the raw data, feature extraction and/or generation, feature selection and feature evaluation [20]. This approach may involve the use of non temporal as well as temporal features [20]. The output of feature construction is a rich feature set that enables computational models' use [20]. Zheng et al. formulated log preprocessing in three integrated steps: event categorization to uniformly classify system events and identify fatal events; event filtering to remove temporal and spatial redundant records, while also preserving necessary failure patterns for failure analysis; causality-related filtering to combine correlated events for filtering through apriori association rule mining [39]. Wang et al. proposed the use of two feature extraction algorithms, Word2vec and Term Frequency-Inverse Document Frequency which are respectively adopted and compared to obtain the log information, and then one deep learning method named Long Short-Term Memory is applied for the anomaly detection [37]. For feature extraction, the skip-gram model of Word2vec could capture more effective semantic information of logs in converting words into vectors expressions for anomaly detection than TF-IDF [24]. The unstructured or semi-structured logs are parsed into structured data and features are extracted before log analysis [24]. Figure 3.5 shows the nature of unstructured logs generated out of our Postgres database server. Several previous works show that they assumed each log contained a timestamp and thread ID or request ID to distinguish from different threads, and converted the unstructured log data into specific keyword formats, and then adopted these keyword sequences and log-related timing information for subsequent anomaly detection [24]. Later, the logs were grouped together by edit distance and the time threshold was set to filter

the duplicate logs. Tuor et al. have modelled the stream of system logs as interleaved user sequences with user-metadata to provide precise context for activity on the network; this allows the model, for example, to identify what is truly typical behavior for the user, employees in the same role, employees on the same project team, etc. [36].

However, based on the goals and objectives of log analysis, the method of preprocessing logs might differ. If the log is completely non-categorical i.e. some kind of range numbers varying from negative infinity to positive infinity, we will not need word-vector or TF-IDF (Term Frequency - Inverse Document Frequency) models and but we may use them if the log data is categorical. Since we are analyzing the database logs, our most frequent encounters will be with categorical log data. The idea of detecting anomaly in the log data by comparing the predicted event with the actual event has been discussed in many research publications. The database logs also contain the query statements executed by several users which is another important attribute in the log data. For predicting the next possible attribute in the query or next query itself in the series, it becomes necessary to adopt natural language processing algorithms.

3.2.3 Rule Based Control and User Pattern Analysis

Detecting some anomalies from the logs may not always require the use of machine learning models. The user pattern analyzers or the machine learning models can make incorrect decisions, maybe due to shorter training period, but rule based access control can act as a compensation for such incorrect classifications [28]. When the data is shared with the trusted third parties, a data license agreement is agreed among the licensor and the licensee. The agreement can specify few conditions, for example only two members of the organization are authorized to access the database. With such specific conditions, the anomaly detection can also be simply achieved using rule based analysis. The rule-based approach is primarily to express expert knowledge as a set of rules that require developers to write in advance using scripts, the operator needs to specify two types of rules, one for regular expressions that extract certain text patterns from log messages, and one for performing simple aggregations on extracted patterns [37]. The rules for extracting patterns can be fetched from the data license agreement and it can be used to detect rule violations from the logs. However, simply rule-based matching will not be applicable if the logs have no rule violations but does

have some suspicious database access patterns. Analyzing patterns of database access by the users can uncover suspicious activities and violation of data license agreement. This approach can be used to detect anomalous patterns, e.g. unusual IP address, access time, and excessive query traffic [28]. Roh et al. propose the specific target items for analyzing user access patterns, they are hourly, weekly, daily and monthly query traffic and user IP address [28]. Putting it mathematically, they have proposed the below equation for detecting anomalous user behaviour.

$$x > u + wq \tag{3.1}$$

Where simply, x is the number of generated queries, u is the average traffic, q is the standard deviation and w is the weight value. If the value of x is greate that u + wq then the system determines queries are anomalous [28].

Another approach of user pattern analysis is by auditing the sequence of logs per user session. Using intent recognition models, we can determine the major goal or intent of a particular user session and identify if there are any suspicious motives. However, it is possible that violations can occur by activities taking place in several different sessions and therefore, rule based and user pattern analysis cannot be the only single method for detecting anomalies from database logs.

3.2.4 Machine Learning Approach

Machine learning models can help identify the direct or indirect data license violations. These models can learn behavior patterns of different users by automatically extracting feature and detect anomalies when log patterns deviate from the trained model. For building a machine learning model for violating or predicting data license violations from the logs, we can use LSTM (Long Short Term Memory) model. LSTM Model can learn behavior patterns of different users by automatically extracting feature and detect anomalies when log patterns deviate from the trained model [38]. LSTM could achieve the best results in anomaly detection of system logs based on the feature extraction methods, especially the Word2vec method [37]. Wang et al. propose a method for anomaly detection that combines natural language processing methods, such as Word2vec and TF-IDF and deep learning algorithms of LSTM, and verify its effectiveness and accuracy with the system logs [37]. Their anomaly detection results show that LSTM performs better than Naive Bayes and GBDT algorithms on both of the two feature extraction methods, demonstrating that LSTM

has strong ability in capturing the contextual semantic information of logs, is insensitive to different features, and will be a powerful and promising tool in system logs anomaly detection analysis.

On the other side, Roh et al. proposes a simple machine learning model which uses Naive Bayes, one-class SVM and one-class Nearest Neighbour. They rely on the use of an anomaly free database where the database log represent the normal user behaviour [28]. Then the classifier is trained with this non-anomalous log data and used to identify anomalous behaviour [28]. The machine learning model simply relies on the features of the queries like query command, query length, projection relation, selected attribute, where attribute, order by attribute, group by attribute and joined tables [28]. Then the classification result is produced by their model. One class SVM had the best performance.

Tuor et al. make use of Deep Neural Networks (DNN) and Recurrent Neural Networks (RNN). The RNN models the temporal behaviour in the log data whereas the DNN model does not. To aid analysts in interpreting system decisions, the model decomposes anomaly scores into a human readable summary of the major factors contributing to the detected anomaly [36]. The focus of the research is on insider threat detection but the underlying model offers a domain agnostic approach to anomaly detection. The LSTM model has the greatest potential to generalize: the model could be applied to individual events / log-lines, using its hidden state as memory to detect anomalous sequences of actions. Since anomaly can take new and different forms, it is not practical to explicitly model it; their system also models normal behavior and uses anomaly as an indicator of potential malicious behavior.

Additionally, model interpretability is vital for administrator and analyst to trust and act on the automated analysis of machine learning models [7]. The work of Brown et al. demonstrate model performance and illustrate model interpretability. Their language model generates all output with a single anomaly score, the negative log-likelihood, for each log-line [7]. They illustrate two approaches to analysis of attention-equipped LSTM language models: 1) Analysis of global model behavior from summary statistics of attention weights, and 2) analysis of particular model decisions from language model predictions and case studies of attention weights.

One more approach towards log surveillance is creating several monitors to keep an eye on the log activities. Leveraging the fact that hiding from multiple, redundant monitors is difficult for an attacker, to identify potential monitor compromise, Thakore et al. combine alerts from different sets of monitors by using Dempster-

Shafer theory, and compare the results to find outliers [35].

However, the main aspect is that it is not only about pursuing the detection of strange events but also to generate a summary of the data processed in order to simplify the human supervision of the logs [24].

3.2.5 Summary

We have seen different methods for log preprocessing, rule based access control, user pattern analysis and machine learning modelling for log anomaly detection. It can be understood that use of any single approach does not prove as a complete solution towards detecting anomalies from the logs. Understanding the main concerns associated with sharing of the private data and studying different methodologies, Roh et al proposed three different models collectively viz. User pattern analysis, Machine learning model and Rule based analysis. After each model generates the result, we will need to aggregate their results using a decision logic proposed by Roh et al. to come up with a final anomaly score [28].

While the data is being used, it is important to understand the intention of each user accessing the database. When a user accesses the database, a new session is created and when the database connection is closed, the session is stopped. All the interesting patterns lie inside this particular time series session and we can model our access pattern analyzer to extract the objectives and intents of the user during the particular session. We should also examine the behaviour of queries, calculate average and standard deviation of query traffic for each target items [28]. The output of this component would generally be a boolean result for any predictable anomalous access patterns.

Secondly, we need a machine learning model to effectively predict if there is any chance of anomaly occurrence in the recent future. Based on the database access patterns of the users, our system must be proactive to detect and report the chance of violation of the data license agreement. Though we can use different LSTM, DNN and RNN models for developing our machine learning model, the simple idea of Roh et al. will also prove equally efficient [28]. The query attributes such as query command, query length, projection relation, selected attribute, where attribute, order by attribute, group by attribute and joined tables are the key attributes to detect the violation of data license inside the logs. A simple one-class SVM classifier can be used to detect the outliers in the logs and our model generates a classification result.

The output of this one-class SVM model would return all the logs that it flags as anomalous.

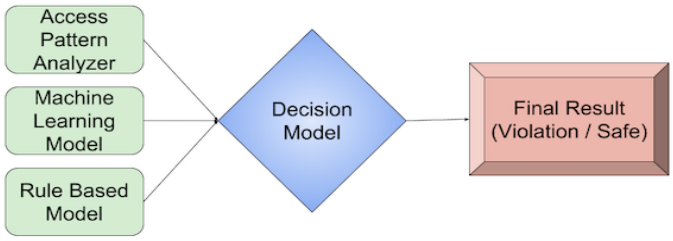

Figure 3.3: Proposed method

Finally, our rule based analyzer will detect the anomalous behaviour in the logs based on the rules created by data license provider. The output of this component would be flagging logs that violate the rules defined in this rule based analyzer. The rules can be derived by mapping data sharing agreement or privacy policies into programmable rules. At times, the violation in the database logs can be easily identifiable with help of regular expressions or pattern matching and use of rule based analyzer is a wise choice in this scenario. Again, since the user pattern analyzer and the machine learning model might be trained using limited amount of data, it is possible that the results provided by each of the former two methods might be less accurate until they are trained very well with huge training data. We can use the rule based analyzer to compensate such incorrect classifications. Our rule can simply contain the following attributes: User, user role, query command, access table, IP address, week, time, day and the dependent variable as "Classification" which contain two classes viz. normal and abnormal.

In this way, we can build a robust model which contain three separate sub-models doing their individual tasks and finally each of their results are combined and calculated as a single result by the Anomaly Decision model.

3.3 Our implementation

Data licence usage monitoring is a business process that monitors licensed data usage on the licensee platform for data that originated on the licensor platform. We investigated the feasibility of using machine learning techniques to examine licensee database log data and compare it with data licence info and simulated licensed data

version metadata. The first phase of the research involved looking for licensed data usage by users not listed on the data licence. This is where the rule based access control helps. Although, not all the data sharing agreement violations could be detected using rule based access control, for example, in the third policy in our data sharing agreement (introduced in chapter 2), it states that no joins could be made to identify an individual based on allergy information. There could be many ways to join multiple tables and know the desired information. This is where machine learning approach could help in balancing the limitations of the rule based access control mechanism. The second phase of the research involved looking for licensed data usage that is not allowed. It required more complicated simulated data that was developed using Synthea - an open source patient data simulator - after challenges with setting up the research test environment using the ML tool to look for unlicensed data use were identified.

3.3.1 Design Challenges

At the beginning, our challenge was to build the tool to offer a solution to the data usage monitoring through database logs using machine learning. The question was "How can we help address the data sharing concerns by monitoring data usage logs?" The tool would greatly impact the entities or industries who have a requirement of sharing the data and yet monitor the usage for ethical purpose. We studied the existing literature to understand the tools and techniques used for log monitoring which we discussed in section 3.1. The most important challenge was to identify the open source tool needed for our application and orchestrating them together. After significant number of revisions, we designed a system architecture for our tool using the ELKF (Elastic-Logstash-Kibana-Filebeat) stack. The figure 3.4 highlights the system architecture of our data usage monitoring tool and the extended work which we carried out later. We introduce each of these components below:

- Elastic-search: It is a distributed, open source search and analytics engine for all types of data. It is built on top of Apache Lucene - a high-performance, full-featured text search engine library.

- Logstash: It is an open server-side data processing pipeline that ingests data from a multiple sources, transforms it, and sends it to a specified destination (in our case elastic search).

- Kibana: It is an open source data visualization dashboard for elastic search.
- Filebeat: It is a log shipper for forwarding and centralizing log data and also supports real time log shipping.

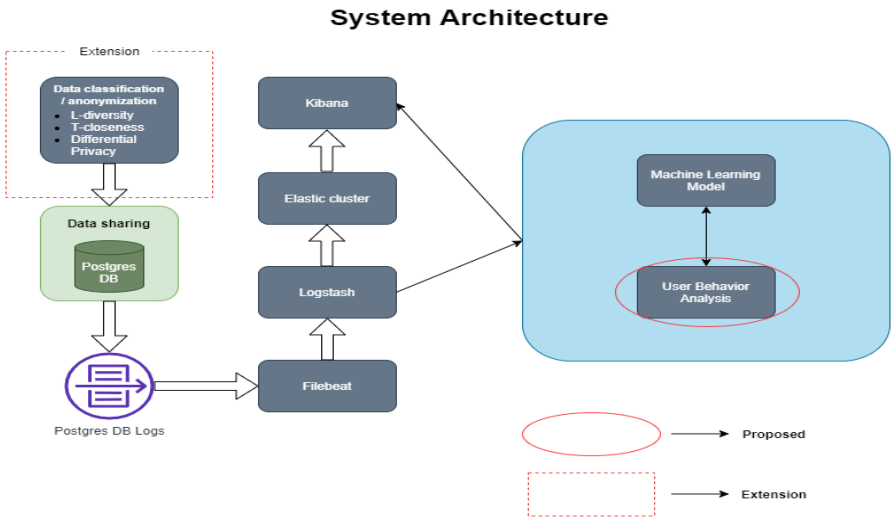

Figure 3.4: System architecture

All of the feature of the above open-source components complements well to our idea of design for this tool. In this chapter we discuss our data usage monitoring tool **without** including the extension module since the extension was not a part of initial requirements. It focuses on data usage monitoring tactic of data privacy quality attribute which we categorized as "Audits and access control". This motivated us further to conduct a survey among the practitioners in the industry to explore additional tactics which they follow in their organization to maintain data privacy which we discuss in chapter 4. The extension of this data usage monitoring tool to provide secure data sharing is discussed in chapter 5.

3.3.2 System architecture

After many revisions, we drafted a suitable system architecture for this project as shown in figure 3.4. It is important to note here that Synthea [3] data mimics the real world health data which health authority may want to share to a third party using data sharing agreement. Based on the overall system architecture of our research partner, there can be one or more databases which would be accessible using cloud

infrastructure. We loaded the Synthea [3] data in a Postgres database server through which the data could be queried. Once the data is shared with mutually agreed upon data sharing agreement, the consumers could issue queries to the data to get desired information. All the interactions with the data is recorded in the database server log. Although, it is expected that the queries issued to the database adhere by the data sharing agreement, there can be attempts to issue queries that violate the data sharing agreement, maybe accidentally or on purpose. For the purpose of this experiment, we ourselves were the data consumers issuing queries to our shared Synthea data. To replicate the real world scenario, the majority of the queries we issued were according to the data sharing agreement and only a few were violations. In total, we issued 3185 queries, out of which 144 violated the data sharing agreement. Our system architecture helps with this log monitoring problem. The key input for this system are Postgres database logs and output is flagging and returning the queries or logs that indicated violation of data agreement. Figure 3.5 highlights the log data format of Postgres database. For this process to function efficiently, we designed the above architecture for our system using the open source Elastic - Logstash - Kibana - Filebeat stack.

Let's understand each of these components with respect to our requirements in brief:

Data classification / anonymization: This is a key module that helps anonymize the data and quantify data privacy before the data is being published. This is an extension to our study and we discuss these techniques in detail in chapter 5.

Data Sharing: This module is responsible to share the private and anonymized data among the data consumers. The data sharing can be done using data sharing agreements or blockchain.

Filebeat: This component is a data shipper for logs. It will directly bring the logs from Postgres database log directory and supply them to Logstash after every specified time intervals.

Logstash: It is used for log data processing and transformation of data as required by the ML model and Elastic cluster.

Machine Learning Model: This component holds the logic for predicting anomalies and detecting suspicious patterns which is trained using one-class SVM clustering approach.

Elastic-cluster: A search engine that indexes the transformed data used for searching everything (anomalies, statistics etc.) about the log data.

Kibana: A data visualization dashboard for visualizing insights.

User Behavior Analysis: This component is expected to be a separate machine learning model that analyzes user behaviour or access patterns through logs. We propose this as the future scope for this research project.

3.3.3 Postgres log structure

For this research, we used Postgres database which acted as a data provider but any SQL db works. The data provider makes a data licensing agreement with the data consumer. This agreement contains the rules of how the data should be accessed and used. The data consumer signs the agreement to use the data by the prescribed rules. Our tool is an attempt to monitor data consumer in order to make sure the agreement is not being violated intentionally or accidentally. As we learnt in the previous section, logs are the best way to monitor the usage of services. For our purpose, Postgres database logs is the key data source. These logs maintain the information about how the data consumer accesses the database in order to consume data. The structure of logs which we receive as input from postgres DB are shown in the Table 4.1.

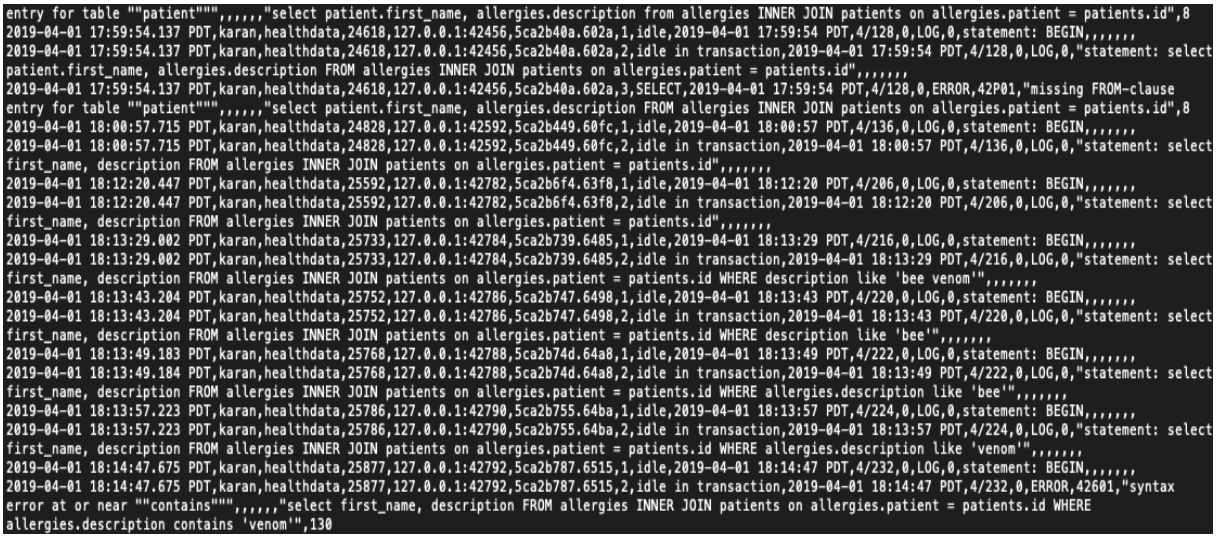

Figure 3.5: Unstructured Postgres logs

3.3.4 Rule encoding phase

Our rule encoding phase transforms the information from these logs into another one-hot encoded database for the purpose of training our machine learning model.

Index	Attribute
0	timestamp(3) with time zone
1	user_name
2	database_name
3	process_id
4	connection_from
5	session_id
6	session_line_num
7	command_tag
8	session_start_time
9	virtual_transaction_id
10	transaction_id
11	error_severity
12	sql_state_code
13	message_text
14	detail
15	hint
16	internal_query
17	internal_query_pos
18	context
19	query
20	query_pos
21	location
22	application_name

Table 3.1: Postgres DB Log Format

The rules are referred from the data license agreement. For the research purposes, we drafted a sample data license agreement and used it to encode the rules in the rule encoding phase. Sample rules assumed for the data license agreement are as follows:
1. Users allowed to use the database: [‘karan’, ‘postgres’]
2. The users should not access the “patients” table
3. The users should not access the patient information using foreign key from allergies, immunizations, observations, encounters or procedures table.

In order to train our ML model, we had to generate sufficient amount of log data by querying the database. We ran the queries against the database to generate the log data. The good queries would be those which follow the rules stated above. The bad queries are those violating the rules stated above. 80% of the queries issued

were the good ones and 20% were those that violated the rules. Once the log data is generated, the rule encoding phase starts its job which generates the ML training dataset.

3.3.5 Machine learning model training

After encoding, our ML dataset consists of the tabular structure shown in table 4.2.

Attribute	Datatype
Permitted_username	Binary [0 = present, 1 = absent]
Permitted_database	Binary
Allowed_tables	Binary
Joined_tables	Binary
Where_attributes	Binary
Group_by_attributes	Binary
Order_by_attributes	Binary
Log_line	Integer

Table 3.2: ML Dataset

For all the attributes, we have binary/boolean value representation except for log line which is an integer representation of the line number of the log in the Postgres log data.

In order to detect violations in the data license agreements, we named these violations as anomalies. We looked at it as a novelty detection problem. Anomalies are something which may not appear frequently. These are rare events that are often disguised among all the normal events. Our log data was imbalanced, so we had more normal events than anomalies which is reasonable in the real world scenario as well. Due to this, supervised learning approaches are less likely to perform reliably. We needed an unsupervised learning solution. As we discussed in section 3.2.5, we decided to go with one-class SVM clustering approach. One-class SVM is an unsupervised learning algorithm which is trained only on the 'normal' data. In this way, the model only knows what are the good patterns within the logs. Whenever an unseen pattern is identified by the model, it flags it as an anomaly. This offered us an optimal solution for our application.

3.3.6 Evaluation

Our training data included the logs which we manually generated by issuing queries to the Synthea [3] database in our Postgres instance. We issued 3185 queries which generated 3185 log lines. Out of the 3185 queries, 144 queries were anomalous. As we discussed, we trained our One-class SVM model using all the non-anomalous data which consisted of 3041 log lines. These logs were converted into one-hot encoded ML dataset as depicted in Table 3.2. After experimenting with different hyper-parameters for one-class SVM model, we were able to come to an optimal set of hyper-parameters that best detected the anomalies using 10-fold cross validation approach. Normally, the evaluation metric of a machine learning model is in terms of accuracy, which is defined as accuracy = number of correct predictions / total number of predictions. Although, there was a significant amount of imbalance in the distribution of classes (anomalous [-1], non-anomalous[1]) and therefore we decided to choose balanced accuracy as the measure to determine the quality of the predictions of our model. Balanced accuracy is defined as Balanced accuracy = (true positive rate (TPR) + true negative rate (TNR)) / 2. The smaller dataset size was the main factor affecting our results, and we received 90% balanced accuracy with the following configuration for sklearn.svm.OneClassSVM:

Parameter	Value
kernel	rbf
gamma	0.001
nu	0.03

Table 3.3: ML Dataset

Limitations

The data usage monitoring tool is built to work only with Postgres logs of the given format. Also, the rules on which the model is trained is based on the three basic rules which we assumed in the case study introduction. This is because we did not have practical exposure to real-world data license agreements. Although, we believe this work is extensible and with its further expansion, it could support other database logs and rules can be customised based upon the data license agreements to train the machine learning model. Additionally, the key problem we identified with our machine learning model was that sometimes the good patterns were also flagged anomalous

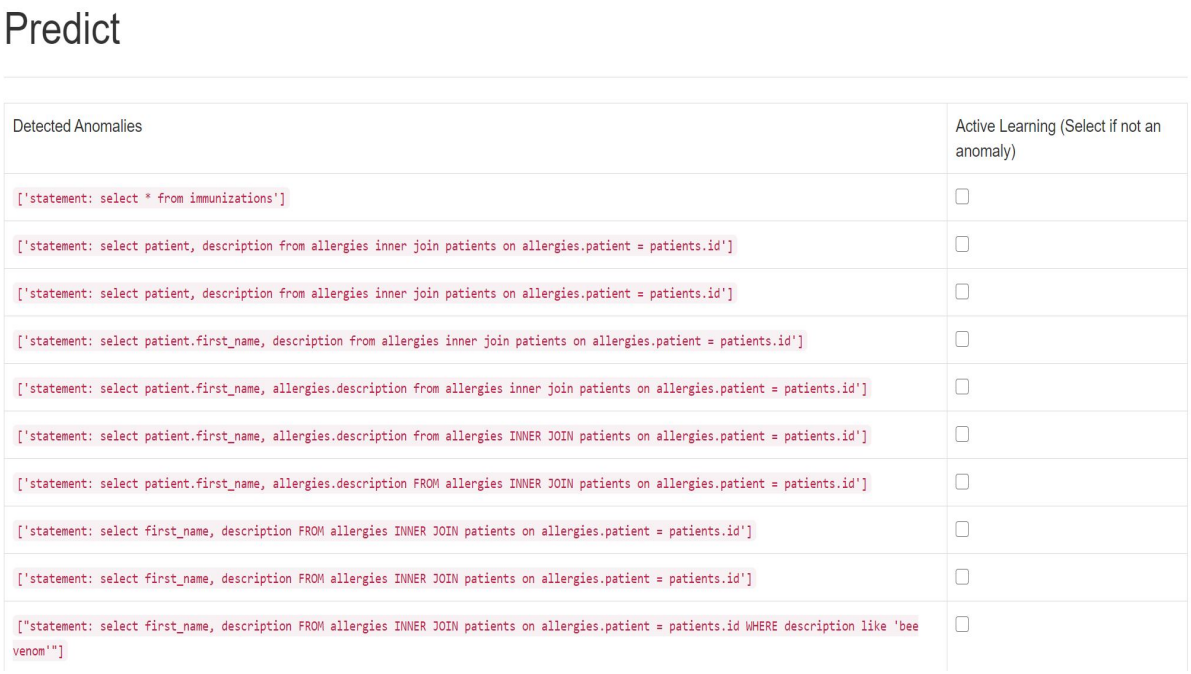

Figure 3.6: Detected anomalies

as the model did not see it before. We decided to tackle this situation using the active learning approach. We built a dashboard that provides an option to the data provider to view all the anomalous activities detected by the model and mark it as normal if it is not an anomaly. This information is supposed to be sent back to the model to re-train and the next time the model identifies such occurrences as normal. Although, this feature is in the prototype phase right now and is a future scope of the tool. Additionally, detecting false positives and using active learning to tackle these false positives is one thing, but there could be instances of false negatives which could totally get neglected if they are not detected as anomalies by the machine learning model. In this case, even if there is a data sharing agreement violation, it may go unnoticed as these false negatives may get ignored by the machine learning model. There is a need to balance this limitation of the machine learning model and the idea of user access pattern analysis through logs may help in balancing such situations as it would help make the system pro-active in terms of identifying data access patterns through logs that may lead to a potential violation of a data sharing agreement in advance.

3.4 Chapter summary

This chapter was focused to find an answer to our RQ 1: "**How machine learning can be used as audits and access control tactic to maintain the quality of data privacy?**". We learnt how machine learning could automate the task of auditing data usage and access control thereby contributing to a key data privacy tactic 'audits and access control' which we introduced in section 1.2. Furthermore, in this chapter we emphasized the tool and methodology we developed for addressing data license monitoring to monitor data access violations as a tactic for data privacy quality attribute. Then we discussed the system architecture of the tool and different modules that work together to form a system. Finally, the chapter concludes by introducing the limitations and future enhancements of this tool. This work was later extended by introducing an extension module to ensure secure data publishing and introduce it as a tactic in maintaining data privacy. We discuss the extension module in detail in chapter 5.

Chapter 4

Industrial survey

4.1 Introduction

In chapter 3, we learned about the specifics of our data usage monitoring tool which we built using the specific set of requirements discussed in chapter 2. At this point, we were keen to know about the ideas behind the tool in the industrial paradigm. Using the design science approach, we decided to conduct an industrial survey among the data driven developers within the industry to know about the tactics they use to ensure data privacy while developing data driven tools. Before conducting the survey, we studied several other similar studies which were conducted in this area and we found that most of them were focused on developer awareness on data privacy [33, 18, 6, 16, 31]. While analyzing the existing studies, we were able to identify a gap between those studies and our work. The existing studies were targeted towards general audience of software developers, which may or may not include data driven developers, although our survey is only targeted towards data driven developers within the industry. In this chapter, we discuss our findings on our RQ 2: "What tactics do the data driven practitioners in the industry follow or suggest to ensure data privacy?" by reviewing the related studies conducted in this area, identifying the gap between the existing studies and our study, and discussing the specifics of our methodology and results of our survey among data driven developers in the industry.

4.2 Related work

4.2.1 Introduction

Data and information helps any organization make efficient and reliable decisions for themselves and their users which is adding tremendous value in the modern life. From search engines to online shopping websites, the way we used to interact with these services has drastically changed over the decade, all thanks to the humongous amount of data being generated every second all around the world. This also means that the large amount of generated data is being collected, stored and processed by the service providing organizations. Moreover, the practice of data collection has been arguable and less transparent to the users. Due to this, the advent of data protection regulation acts like GDPR (General Data Protection Regulation) act, PDPA (Personal Data Protection Act) etc. was eminent. Even though these acts are in place and also the privacy policies of respective organizations where they mention their data collection practices, it is another challenge in software engineering to make the software algorithm follow the context of the privacy policies.

4.2.2 Developer and user viewpoint on data privacy

Sheth et. al. [33] conducted a study to explore the privacy requirements for users and developers in modern software systems, such as Amazon and Facebook, that collect and store data about the user. Their study consisted of 408 valid responses representing a broad spectrum of respondents: people with and without software development experience and people from North America, Europe, and Asia. While the broad majority of respondents (more than 91%) agreed about the importance of privacy as a main issue for modern software systems, there was disagreement concerning the concrete importance of different privacy concerns and the measures to address them. The biggest concerns about privacy were data breaches and data sharing. Users were more concerned about data aggregation and data distortion than developers. As far as mitigating privacy concerns, there was little consensus on the best measure among users. In terms of data criticality, respondents rated content of documents and personal data as most critical versus metadata and interaction data as least critical [33].

The new European General Data Protection Regulations (GDPR) that came into effect in 2018 has generated considerable interest towards privacy design guidelines in

software system designers, such as the Privacy by Design (PbD) principles. Privacy by design principles suggests for privacy to be taken into account throughout the whole software engineering process. Lack of understanding on the privacy risk perceived by users could result in systems that do not cater for user privacy expectations and invade user privacy when users interact with those systems [31]. Senarath et al. [31] developed a measurement to calculate the privacy risk perceived by users when they disclose data into software systems. While it has been widely accepted that the relatedness of data affects the privacy risk perceived by users when they disclose data into software systems, so far there is no evidence as to how related a data item should be in order to make users feel comfortable sharing those data into the system. That is in a system design, after measuring the privacy risk perceived by users against the data that is used in the system, developers could reduce the visibility of data with high privacy risk. Their model provided a cost effective alternative for developers to approximate the privacy risk perceived by users when they design software systems.

4.2.3 Privacy education

As the Internet becomes more technically complex and, at the same time, more intertwined with everyday life and the well being of organizations, we face the question of how to educate users to help them protect their privacy. Kang et al. [18] conducted a qualitative study to investigate users' mental models of the Internet and their knowledge of data flow on the Internet. They examined how they conceptualize the process of connecting to the Internet and how they think others can access their data online. Analysis revealed strong differences among users with different educational backgrounds. The majority of those without computer science education had simple, service-oriented mental models whereas those with a background in computer science had an articulated many-layer model of the Internet that included key entities and organizations. People with a more articulated model expressed higher awareness of specifically who might have access to their personal data and communications. Yet technical background was not directly associated with more secure behavior online. Almost universally, participants' privacy protective actions or lack of action were informed by personal context and experiences, such as a feeling they had nothing to hide, and in some cases by immediate cues in the online environment such as a security emblem or famous company name. Their work suggests a need for more research into privacy protections that reduce the responsibility on users to understand how the

Internet works and to make myriads of privacy protection decisions based on their technical knowledge.

4.2.4 Organizational climate

Ayalon et al. [6] came up with a study whose aim was to understand how personal and working environment features affect developers' professional privacy attitudes and practices by surveying developers. Understanding these effects will help to understand how privacy controls and user interactions are designed, as well as providing some guidelines for enhancing privacy in information systems design, which can have a significant effect on the developers' communities. Their study points to the role of developers in forming organizational privacy practices and to their place in affecting the dynamics of privacy. It can be learnt that the legal background is less impactful than the organizational privacy climate. A possible explanation of this finding is that the climate mediates the legal and business environments in which the organization operates. This finding is important because it highlights the informal aspects of organizational privacy conduct. In this sense, privacy is similar to other quality characteristics of organizations, much like work safety, in which the climate was found to affect employees' safety behavior much more than formal policies [16].

Senarath et al. conducted a research which attempted to identify the issues faced by software developers when they attempt to embed privacy into software applications. Based on the findings of the study they derived guidelines to effectively support software developers when they attempt to embed privacy into software applications. However, the relatively small sample of participants in the study should be taken into account in generalizing their findings. Their findings indicate that developers have practical issues when they attempt to embed privacy into software applications [32]. Developers find it difficult to relate privacy requirements into engineering techniques and they lack knowledge on formally established privacy concepts such as PbD, which are well known in the domain of privacy research. Because of this the solutions researchers implement, expecting developers to be well versed with the privacy concepts may not work in software development environments. When developers lacked knowledge, their personal opinions and complex system requirements seem to take precedence over privacy requirements which eventually result in software applications with limited or no privacy embedded. They suggest that developers should be given formal education on privacy practices.

Cavoukian et al. [9] shows an excellent example of how enhanced privacy accountability and assurance can be achieved within an organization by applying Privacy by Design principles, in a thoroughgoing manner. So imperative today are the goals of enhanced accountability and assurance, so universal are the PbD principles, and so diverse are the contexts within which these principles may be applied, that the future of privacy in the 21st century information age may be limited only by our collective imagination and will. There are virtually infinite ways by which organizations can creatively "build privacy in" to their operations and products, to earn the confidence and trust of customers, business partners and oversight bodies alike, and to be leaders in the global marketplace [9].

4.2.5 Challenges to embed data privacy in the software

It is clear from the previous subsection that in many cases the organizations' privacy policies and their broader privacy climate are not always aligned [17]. In some organizations, the organizational climate allows, and even promotes, behavior that is inconsistent with the official, defined policy or regulations, despite the risks of future losses in terms of money and reputation. Yet, in other organizations, the organizational climate promotes its privacy policy; supervision, communication and educational measures are taken to ensure that employees are aware of, and adhere with, the organizational privacy policy.

Examining the point of view of software developers, it seems that, except in the context of specific domains, software developers are actively discouraged from making informational privacy a priority, being expected to conform to norms and practices dictated by a negative organizational privacy climate [17]. But the problem goes deeper than mere prioritization; many developers do not have sufficient knowledge and understanding of the concept of informational privacy (data protection), nor do they sufficiently know how to develop privacy preserving technologies [17]. If Privacy by Design (PbD) is ever to become a viable practice, a considerable change is to be made for preparing the field for the wide implementation of this policy. The findings of this study suggest that organizational privacy climate highly influences developers' privacy interpretation and behavior; thus, it may potentially serve as an effective mechanism to bring about the required change in the privacy mindset and practices as to informational privacy, starting with the adaptation of organizational policy to the principles of FIPPs (Fair Information Practices Principles) and followed by the

diffusion of this policy into the organizational climate. A well-designed educational program would increase developers' knowledge and skills for designing privacy. Providing developers with knowledge, by means of education, as well as motivation, by means of positive organizational privacy climate, could potentially create the mindset required for designing privacy preserving solutions. Their future research may examine these and other means and their actual effect on developers' perceptions and attitudes toward informational privacy. If successful, this would be an important and necessary step toward wide and effective implementation of PbD.

Oetzel et al. [26] discussed about the four-step Privacy Impact Assessment (PIA) process. It consists of: (1) describe the system landscape, (2) identify privacy risks, (3) mitigate those risks through appropriate controls, and (4) document the analysis and residual risks in a PIA report. This four-step methodology has been called a 'landmark for privacy-by-design' by Ontario's data protection authorities, who invented the concept of privacy-by-design [26]. They reviewed existing security risk processes to inform the creation of a new PIA methodology. Their major contribution was the development of a new set of artefacts. These artefacts help practitioners and researchers understand the relevant privacy regulation landscape and analyse and assess privacy issues by using a systematic step-by-step process. The PIA methodology helps practitioners realise the concept of privacy-by-design in their system development lifecycle. Specifically, the artefacts provide systematic support for representing privacy requirements in the form of privacy targets, evaluating how much protection these targets require and systematically identifying threats and adequate controls. The proposed privacy targets have been systematically derived from legal data protection requirements and privacy principles. Their PIA methodology is built on prior risk assessment experiences and research, especially in the security risk assessment area. The methodology can be verified because each step of the PIA process produces an artefact. Since this methodology will be applied in varied contexts, they expect the artefacts to vary as well [26].

Privacy and data protection constitute core values of individuals and of democratic societies. This has been acknowledged by the European Convention on Human Rights and the Universal Declaration of Human Rights that enshrine privacy as a fundamental right [10]. With the progress in the field of information and communication technologies, and especially due to the decrease in calculation and storage costs, new challenges to privacy and data protection have emerged. There have been decades of debate on how those values and legal obligations can be embedded

into systems, preferably from the very beginning of the design process. One important element in this endeavour are technical mechanisms, most prominently so-called Privacy-Enhancing Technologies (PETs), e.g. encryption, protocols for anonymous communications, attribute based credentials and private search of databases. Their effectiveness has been demonstrated by researchers and in pilot implementations. However, apart from a few exceptions, e.g., encryption became widely used, PETs have not become a standard and widely used component in system design. Furthermore, for unfolding their full benefit for privacy and data protection, PETs need to be rooted in a data governance strategy to be applied in practice. The term "Privacy by Design", or its variation "Data Protection by Design", has been coined as a development method for privacy-friendly systems and services, thereby going beyond mere technical solutions and addressing organisational procedures and business models as well. Although the concept has found its way into legislation as the proposed European General Data Protection Regulation, its concrete implementation remains unclear at the present moment [10].

4.2.6 Gap we address between the literature and our study

In the related work, we studied the recent literature to learn developer and user viewpoint on data privacy. Most of the related work were survey studies among the developers and the users. We get to learn about developer and user perception of data privacy and how organisational climate promotes behaviour that is inconsistent with the defined policy or regulations. However, we do not learn how data privacy is particularly addressed in machine learning or data driven systems by the data driven developers. We made an attempt to identify a gap between the related work and our study. All the surveys and interviews conducted in the literature we studied had a general audience of software developers. It is possible that the respondents may involve machine learning/data driven developers, but these studies were not particularly focused on them. We address this gap in our research. In our survey study, we only targeted the machine learning/data driven developers in the industry who deal with data related operations on day to day basis.

4.3 Survey on data privacy as a quality attribute in the industry

In section 4.1 we learnt that developers are sometimes discouraged from prioritizing privacy due to organizational climate and may not value privacy as much as users do. We conducted a survey among the data driven developers, team leads, architects and product owners in the industry in order to understand their perception and tactics to handle some of the key data privacy practices during the development of the software product. On the contrary to that context, we found that machine learning developers are more aware of data privacy concerns. Although the organizational climate and resource availability still influences their work, but it turned out that they were aware and willing to embed privacy practices in their work. The main intent of this survey was to find an answer to our RQ2: "Which tactics do the data driven practitioners in the industry follow or suggest to ensure data privacy?". The participants in the study belonged to different industries and had varied work titles. The table 4.1 shows the participant demographics.

4.3.1 Methodology and Demographics

The survey was structured in a way that contained both, multiple choice responses and open ended questions. The survey was designed after a thorough literature review of the similar studies in the past and also is inspired from the future research direction of [30]. Our method of analysing the survey responses involved going through the multiple choice responses and encoding the open-ended responses into respective categories [29]. These categories were constructed with the motive of extracting themes using thematic analysis from the open-ended responses to see if we could find something worth noticing beyond the scope of the survey questions. All the open-ended responses were grouped together and all of them were analysed equally regardless of their question groups in our survey. The grouping of responses were done by two different research practitioners. After the grouping based on the theme of our research question was completed, they were openly discussed between the two research practitioners to validate the categorization. After the discussion, the response categorizations were finalized. The analysis and categorisation of these responses gave useful insights and similarities between responses of different groups of the people from different roles and industries. After coding the survey responses, we categorized

the responses in the 6 categories as shown in Figure 4.5, and we identify 4 out of these 6 categories as the tactics followed by data practitioners which we discuss in section 4.3.2.

Id	Role	Exp. in Org. (Years)	Overall Exp. in ML (Years)	Industry	Sector
P1	Product Head	>3	<1	Software Services	Private
P2	Data Scientist	>1	1-2	Insurance, Healthcare , Banking / Finance	Public
P3	Software Engineer	>1	1-2	Banking / Finance	Private
P4	Principal Architect AI / ML	>1	3-5	Software Services, Legal Consulting / Law	Public
P5	Data Engineer	>1	1-2	Insurance	Public
P6	Business Intelligence Manager	<1	3-5	Healthcare	Public
P7	Data Scientist	<1	<1	Insurance	Public
P8	Software Engineer	<1	<1	Software Services	Private
P9	Data Engineer	>1	<1	Insurance	Public
P10	Chief Technical Officer (CTO)	<1	3-5	Software Services	Private
P11	Data Engineer	>1	1-2	Insurance	Public
P12	Data Analyst	>3	<1	Banking / Finance	Private
P13	Data Engineer	>1	>5	Digital Marketing / Consulting	Private
P14	Product Designer	>10	3-5	Consulting	Private
P15	Consultant	>1	1-2	Consulting	Private

Table 4.1: Participant Role and Experience

As our research was specifically targeted towards data driven developers, we collected a few additional demographics with respect to their work environment, such as which programming languages they use, use of PII at work, privacy regulations followed and use of personal data for ml training. Knowing these facts helps us to identify that majority of our participants were data driven developers. All of these responses were either multiple choice or yes/no questions. In the figure 3.1, we get to

see that majority of the participants in our survey use Python as their programming language at work for data driven operations while R being the second popular.

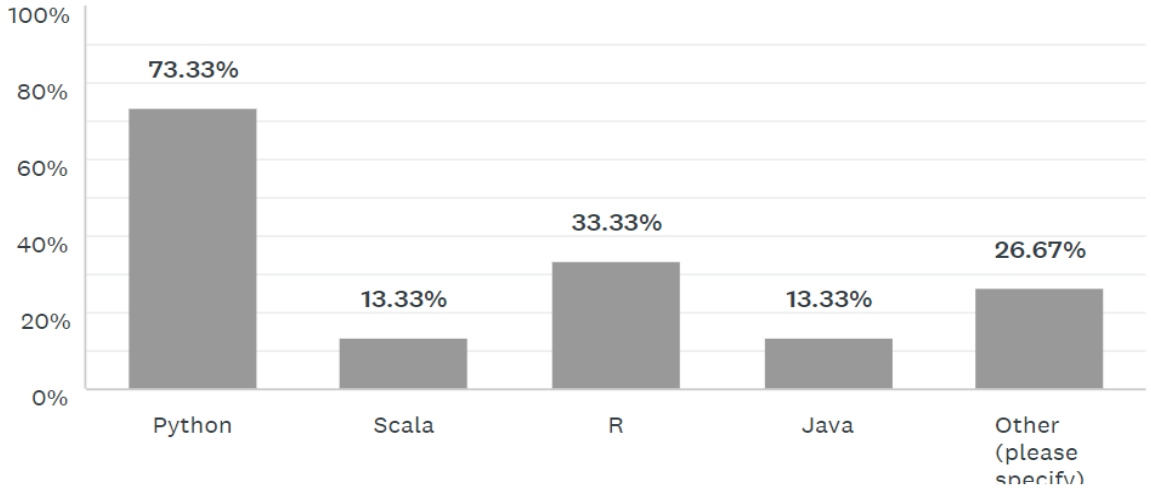

Figure 4.1: Use of programming languages

When we asked this question to our participants "Do you train your ML models using the user's data for purpose of recommendation, improving user experience etc.?" it was known that more than 85% of the participants used user's data for training their ML models for the said purpose. Figure 3.2 depicts those results.

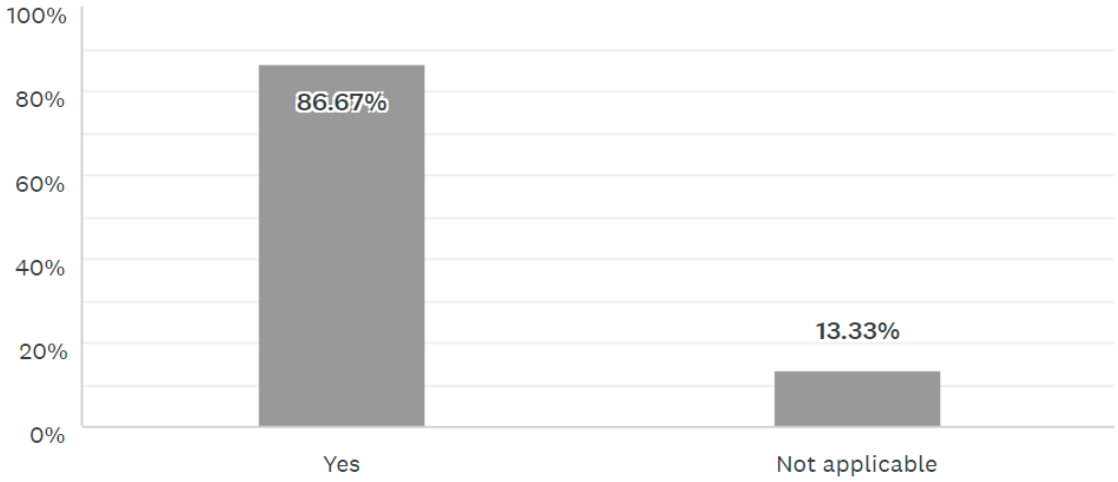

Figure 4.2: Do you use user's data for ML training?

Out of the many third party machine learning development libraries available,

scikit-learn is the most popular one among our participants, as seen in figure 3.3, which is another fact which demonstrates that majority of our participants are machine learning developers. Since its the most popular library among the developers, there is a scope of further research into embedding privacy standards in the library itself or building a wrapper around it.

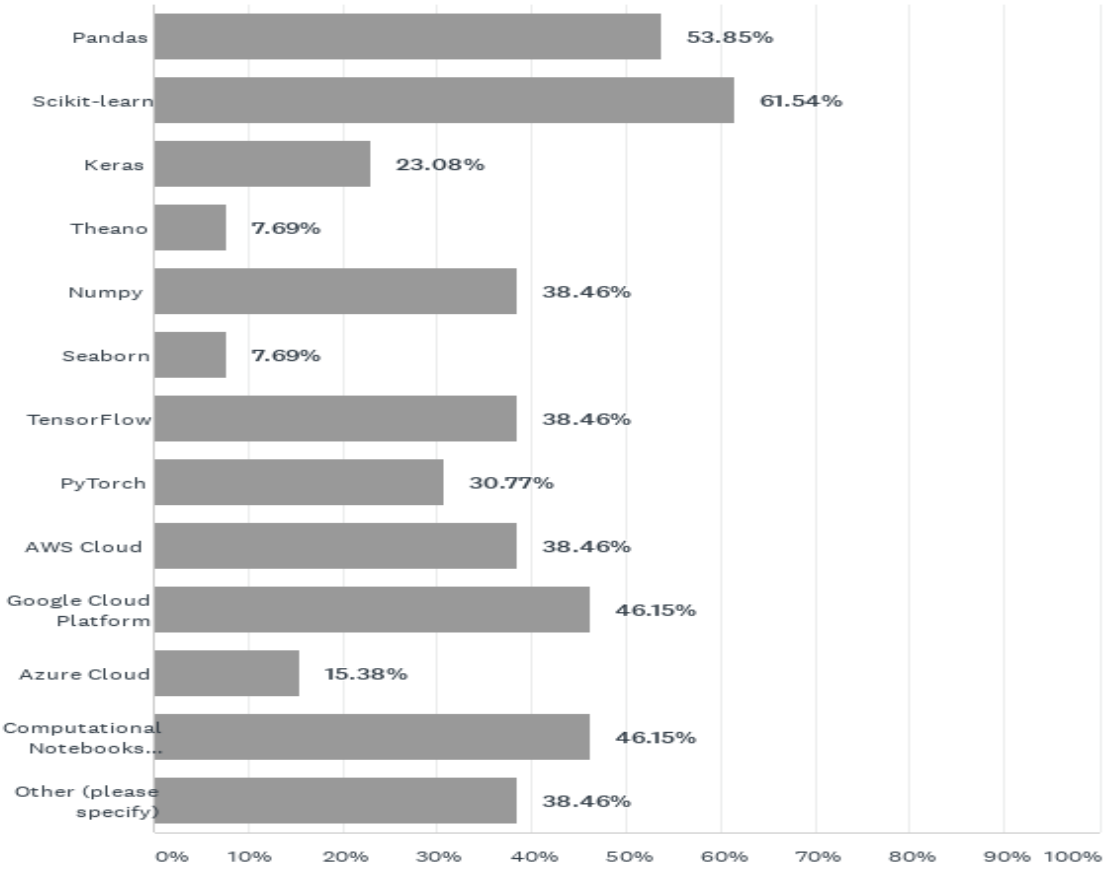

Figure 4.3: Use of third party tools

The majority of our participants had to follow GDPR regulations at their workplace. Most of our respondents belonged to Canada which influenced the response count of use of BC FIPA. Besides, participants involved in health data related operations indicated the use of HIPAA regulations at work.

4.3.2 Categorization of participant responses

In section 4.2, we learned different factors that affect the quality of maintaining data privacy in an organization. We learned that software developers are actively discour-

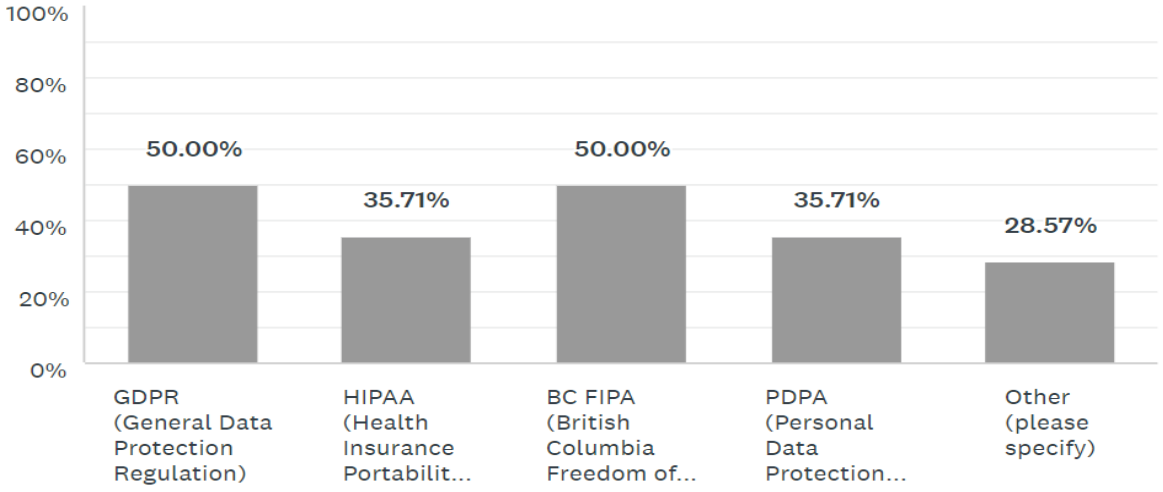

Figure 4.4: Adoption of privacy regulations at work

aged from making informational privacy a priority. Going beyond, many developers do not have sufficient knowledge and understanding of the concept of informational privacy and they do not sufficiently know how to develop privacy preserving technologies. However, in our survey among the data driven developers from startups to large organizations, we found majority of the respondents to be prioritizing privacy over convenience. All the responses that indicated privacy is or should be prioritized above development or user convenience are in this category. Majority of our participants indicated that data privacy is considered as a top most concern in their organization. A few practice removal of PII from data before using it for development or training models. There was also a mention of having user consent for data collection and not using their data for development without their consent. One of participants in another leading global software consulting firm mentioned that they are working towards encrypting data before sharing it or their using it to train their machine learning models. It is a positive sign that majority of our industry participants believe that user data privacy should be a priority above any thing. Although they did mention that there is a still a need to establish formal and well defined practices to achieve this. Many of the participants were working towards achieving this in their organization.

The purpose of this survey was to answer our key research question which is "Which tactics do the data driven practitioners in the industry follow or suggest to ensure data privacy?". Section 4.3.1 describes our methodology which we used to

encode survey responses into the 6 categories as shown in figure 4.5. Out of the 6 categories, we identify four of them as tactics followed by the data driven practitioners. Lets discuss each of four these tactics below:

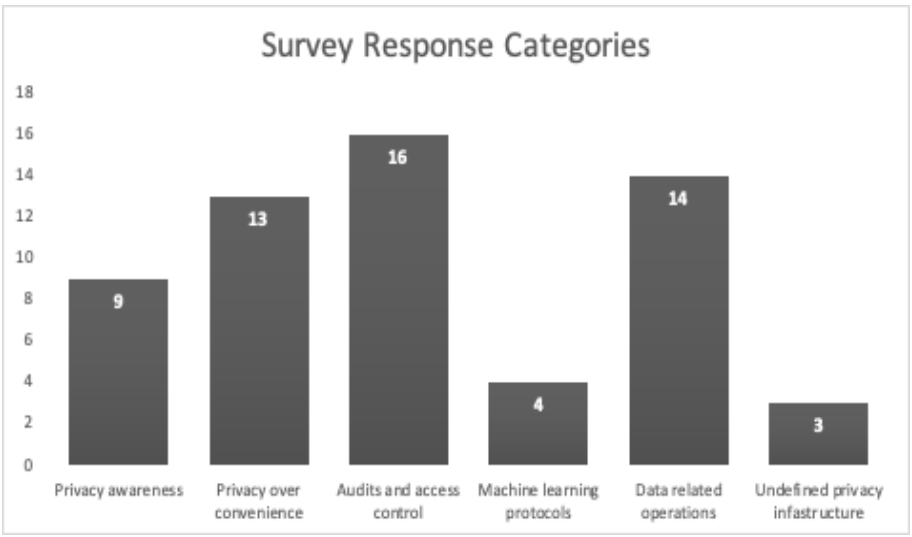

Figure 4.5: Survey response counts per categorization

Audits and access control

We grouped together all the open ended responses that indicated use of access control mechanisms to access data or intended to use it within their respective organization. For example, we asked "What measures do you have in place to ensure data handling practices in your organization strictly adheres to the privacy policy of your product/service?". Some participants indicated through their responses that they have role based access of data assets or they intend to have one. A few also mentioned that they maintain different access control rules for staging and production environments. Participant (P1) of one of the leading organizations in software services indicated that a third party data security organization conducts frequent audits for ensuring data privacy. One of the participants in the public sector organization (P2) indicated that they maintain access control based on geographical jurisdiction. Furthermore, another participant (P15) from a leading global consulting firm indicated that they intend to have a common data sharing platform for all the developers enforced with access controls. It can be seen that majority of the open ended responses fell into the category of "Audits and access control". Many of our participants indicated they have or would like role based access control mechanisms to monitor data

access. A few also mentioned that it is hard to define a fixed standard to monitor data privacy, therefore the manager of the project regulates the privacy guidelines and has the responsibility to manage it manually. This demonstrates an opportunity for the application of our tool in the industry which we developed in chapter 3. Using this tool, we attempt to show how log data monitoring using our methodology can be useful to detect accidental or on-purpose violation of data access control or data sharing agreements.

Data related operations

Responses that indicated efforts or intentions in the direction of data minimization, anonymzation, aggregation etc. come into this category. This was one of the motivations to develop an extension of our tool which we discussed in chapter 3 to support anonymized and controlled data sharing which we detail in chapter 5. Participant (P6) mentioned that they currently aggregate the data to make it ready for analysis purpose removing personal data from the data. Although, they also mentioned one of the main challenges of such data distribution practices to be data re-identification attempts. An entity can easily be identified if the aggregated data is a small number exposing high chance to identify a particular individual. The participant indicated the need for having a quantifiable measure for data privacy and having some control over data sharing which would prevent expose of such data attributes. We discuss a few key techniques available in the literature which could be used for such applications in Chapter 5. These techniques would help to quantify data privacy thereby giving control over data publishing to a certain extent. Another participant indicated that data should be categorized into Personally Identifiable Information (PII) and non-PII. With this we could separate the sensitive and non-sensitive data making it easy for the use of software development and business. Data anonymization was another most discussed aspect which the participants indicated, although majority of the participants said that removing the personal information from the data is the only way they are aware of achieving anonymization.

Privacy awareness

We discussed in section 4.1.2 that developers lack the knowledge for effectively designing privacy preserving technologies and a well-designed education program would increase awareness about this among the developers. This would potentially create

the mindset required for designing privacy preserving solutions, which would also enable effective implementation of Privacy by Design (PbD). In our responses, we found the participants discussing similar ideas. The responses related to good work ethics, data privacy challenges, test cases for data privacy, privacy protocols etc. come into this category. A few participants indicated there is a need to educate both, the developers and the users about data privacy. Most of the times, even at work, developers indicated that they need formal education on privacy standards. From the responses we received, it could be inferred that the privacy standards and requirements are not clear enough for the developers to understand and implement at work. One of the participants mentioned that following a privacy standard is a challenge in their organization because they are not clearly defined and not measurable. One of the respondents said that there should be test cases in place to quantify data privacy and it should be automated than manual. Another participant indicated that they have a test case which attempts to identify individual from their existing data, but attempts to reduce the bias nature of the tests were not discussed. Another interesting response was to assign each data application some sort of internal certification. For example, red data (confidential data not even for private use), orange data (private data, limited to those with a business need-to-know), yellow data (non-public data which shouldn't be disclosed), greed data (publicly shareable). The participant mentioned that they have a architecture for threat modelling of the data and they conduct reviews by a security expert in their organization.

Machine learning protocols

Responses that indicated the use of machine learning for monitoring and achieving data privacy are in this category. Participants indicated that identifying PII within the data is important. Having a reliable machine learning model would make the task quicker and efficient. One of our questions was "how do you ensure that your ML model forgets that data when users request their data deletion as per data regulations?", many participants viewed this as a challenge. Although one of our respondents mentioned that having a data reproducibility mechanism to trace back to the original data using which the ML model was trained will prove useful. Many of our participants mentioned that they remove all the PII before training the model, although we had one another response which mentioned that the definition of PII changes from project to project in their organization. This again encourages the need

to have more clear privacy standards and awareness about privacy impact assessment.

4.3.3 Validity Threats

This section discusses some of the potential threats to validity of this study.

Number of participants

We attempted to reach out to 19 participants for our survey and had 15 respondents in total. Most of them belonged to higher and decision making positions within their organisation making our study qualitative. We tried to reach out to a broader audience in terms of industry sector, and also tried to manage the number of respondents from each industry section. Yet, we believe that in the future having more participants may help in gaining additional perspectives from several other industry sectors and make the analysis more robust.

Data categorization

The data extraction and thematic analysis was done using the method we described in section 4.3.1. The categorization of the responses into the respective categories as shown in figure 4.5 was done based on the conclusions of two qualified research practitioners in this study. Although, given more time and involvement of more practitioners and participants would improve the data categorization as we can have more themes to analyze.

4.4 Chapter summary

In an attempt to explore answer to our RQ 2: "Which tactics do the data driven practitioners in the industry follow or suggest to ensure data privacy?", this chapter goes from explaining the challenges to embed data privacy in a software in the related work to conducting an industrial survey and then knowing the tactics followed by the data driven practitioners in the industry to address challenges. The thematic analysis of the survey and response categorization of the participants helped us to know the tactics which the data driven developers follow in the industry, which helps to answer our RQ 2 as we were able to extract four key data privacy tactics in software engineering followed by the data driven practitioners in the industry, which

we discussed in section 4.3.2. Although, as discussed in the limitations, we could know about more tactics and specifics by further conducting the survey with more number participants. The main intent of conducting this survey along with RQ 2 was also to explore the applicability of our tool in the industrial paradigm which we discuss in chapter 3. We learn through the responses of our participants that audits and access control is the most preferred tactic among data driven developers which gives an important scope to our data access monitoring tool. Furthermore, after being motivated from the responses of our participants, we also decided to focus on "data related operations" tactic and began working on developing an extension to our tool to support secure data sharing. In the next chapter, we discuss in detail about this extension of our tool and different algorithms we used to support secure data sharing.

Chapter 5

Secure data publishing techniques

5.1 Introduction

This chapter mainly discusses our findings for our third RQ: **"Which techniques could be used to secure sensitive information before data publishing to gain better control over data sharing?"**. From developing a data access monitoring tool to taking an design science approach by conducting an industrial survey to learn about the applicability of the tool in the real-world setting, we gained important insights on tactics followed by industry professionals to approach privacy preserving data science through software engineering. In chapter 4, we learned that audits and access control tactic was the most popular among our participants. After realizing the applicability of our tool which we described in chapter 3 for audits and access control tactic, we were further motivated to explore the second most discussed topic by our participants, which was data related operations. Any re-identification on an individual through a public dataset may potentially harm study participants as it will release individual's personal information into public. To address such issues, privacy techniques such as k-anonymity, l-diversity and t-closeness were introduced [22]. This motivation led us to design an extension on top of our existing tool which focused on reducing the data privacy related concerns while sharing the data itself. We primarily focused on data anonymization through L-diversity [25] and T-closeness [19] and implemented a most recent and popular notion of controlled data publishing through data aggregation through differential privacy [13]. The concept of differential privacy was introduced about a decade ago [13]. It was a major step in disclosure avoidance of the citizens of the nation and have a control over secure sharing of census data

by providing microdata to the public instead of the original aggregated information. Microdata are usually composed of individual records containing information collected on persons and households. The unit of observation is usually the individual, but can be the household, family, etc. The responses of each person to the different census questions are recorded in separate variables. Microdata stand in contrast to more familiar summary or aggregate data. Aggregate data are compiled statistics, such as a table of marital status by sex for some locality. Microdata are inherently flexible. One need not depend on published statistics from a census that compiled the data in a certain way, if at all. Users can generate their own statistics from the data in any manner desired, including individual-level multivariate analyses [2]. The differential privacy approach became more popular when the government of USA became the first to implement this approach on the large scale to release the census data in 2020 in order to protect small populations. Additionally, Google recently published community mobility report data which they use for their Google Maps product using differential privacy to help make critical decisions combat COVID-19 without exposing personal information of individuals.

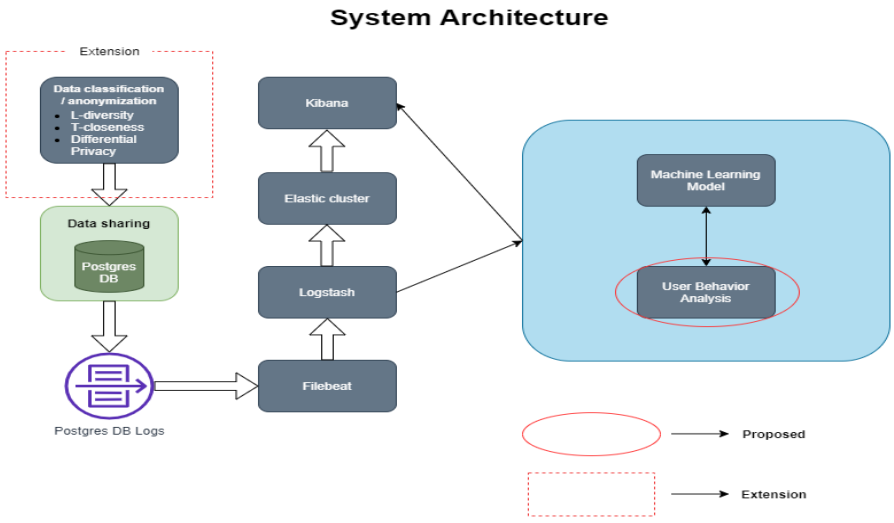

Figure 5.1: System architecture extension - upper left

Figure 5.1 shows the placement of our extended module in our data usage monitoring tool system architecture. This extension conceptualizes our idea of supporting end-to-end data publishing and monitoring through anonymization and controlled aggregated data publishing and then monitoring the access of this published data through our data monitoring tool. After this point, we will be discussing the tech-

niques we used to sanitize the personal information making the data ready to be published as a secure data publishing tactic namely, L-diversity, T-closeness and differential privacy respectively.

5.2 ℓ -diversity and T-closeness

5.2.1 Introduction

The k-anonymity approach ensures that the set of records are indistinguishable from at least k-1 other records with respect to quasi identifiers and these k records form an **equivalence class** [34]. ℓ-diversity and t-closeness are extension to k-anonymity, therefore in the beginning we thoroughly studied the concepts of k-anonymity. K-anonymity specifies that at-least k similar records must exist in the table that share similar quasi identifiers [34]. Later, it was known that k-anonymity cannot prevent **attribute disclosure** [25]. The situation where the adversary can infer some sensitive information about an individual without identifying an individual‘s record in a published data set is called attribute disclosure [25]. For example, if everyone in an equivalence class has HIV, then the identity masking doesn't protect against an attacker realising that somebody has HIV. This problem of attribute disclosure can be addressed with the help of ℓ-diversity approach which states that each equivalence class has at least ℓ well-represented values for each sensitive attribute. ℓ-diversity also has its own limitations, for example, it is sometimes neither necessary nor sufficient to prevent attribute disclosure attacks [19]. This limitation is addressed by a novel privacy notion of t-closeness. It requires that the distribution of a sensitive attribute in any equivalence class is close to the distribution of the attribute in the overall table.

5.2.2 Objective

The goal of this experiment was to implement the discussed data privacy techniques, ℓ-diversity and t-closeness on our sample dataset. The data source is the same synthea dataset which we introduced in section 1.5. Again, we are doing this in order to anonymize and sanitize the personal information within data before publishing it to the data consumers. Initially, there was some data preprocessing involved which includes:

- Converting healthcare expenses and healthcare coverage to ranges
- Inferring age from birthdate and converting to ranges
- Inferring boolean 'died?' from death date
- Anonymizing first and last name
- Inferring gender from prefix

The implementation of this preprocessing on the given dataset can be found in **Preprocessing.ipynb** file which we provide in our replication package.

5.2.3 Approach for ℓ-diversity

As we discussed in section 5.2.1, the dataset needs to be k-anonymized before implementing ℓ-diversity and t-closeness. The k-anonymity approach followed for data anonymization is by adopting the methods suggested by El Emam et al [14]. The ℓ-diversity approach is adopted from Machanavajjhala et al [22]. The given data source had a significant amount of unique personally identifying attributes. These were:

- First Name
- Last Name
- SSN
- Passport
- Longitude

There were also a few quasi identifiers in the data set:

- Age
- Died
- Description
- Gender

It is always a good practice to completely mask the uniquely identifying attributes with random information or hiding it with random characters. In this implementation, this has been achieved by masking such information with an asterisk (*). However, masking a lot of information can lead to data being unusable for analysis. The first and last names were masked where only first letters of first and last names were visible, the rest of the letters were masked with (*). There are also quasi identifiers in the dataset which can help to reveal the person's information indirectly. In our case, they are age, died, description and gender. The approach to achieve ℓ-diversity is shown in figure 5.2. For this experiment, we decided to use the suppression [22] method which removes all the equivalence classes where the distribution of sensitive attributes is greater than or equal to 3 i.e. $\ell = 3$ in figure 5.2. The decision of identifying sensitive attributes solely depends on the data curator. Figure 5.2 shows illustrates the function which we built to ℓ-diversify a given dataset. The function takes three main arguments, 'df' is the original dataset or data table, 'sensitive_col' is the name of the sensitive attribute in the original dataset or data table and 'l' is the maximum distribution of sensitive values in the given 'sensitive_col'.

```
def l_diversify(df, sensitive_col, l):
    for count in df['eq_class'].unique():
        # Count number of unique values in sensitive attribute
        t = len(df[df['eq_class'] == count][sensitive_col].unique())
        if t < l:
            df = df.loc[df['eq_class'] != count]
    return df
l_diversified_df = l_diversify(df, 'description', 3)
```

Figure 5.2: ℓ-diversity python code

5.2.4 Approach for t-closeness

The concept of t-closeness was implemented by adopting the approach of Ninghui Li et al [19]. ℓ-diversity by itself is not sufficient to prevent attribute disclosure attacks [19]. T-closeness requires that the distribution of a sensitive attribute in any **equivalence class** is close to the distribution of the attribute in the overall table. As we learnt in section 5.2.1, an equivalence class is a set of identical quasi-identifying attributes which result from k-anonymity. Therefore, after implementing

ℓ-diversity, another challenge is to decide the threshold value for t-closeness. The t-closeness value is computed by calculating the distance between the local distribution of sensitive values within equivalence class and the global distribution of those values. We used "Bhattacharya Distance Formula" to calculate distance between local and global distribution. The threshold value applies to every single equivalence class and it means that all the records below the specified threshold are assumed safe to be published and records falling above threshold value are vulnerable for attribute disclosure attacks. The best method for doing this would be to analyze the data loss for each different threshold value. In order to determine this, we took the help of bar chart visualizations of records under each equivalence class against the threshold as shown in figure 5.3. The x-axis represents the t-closeness value of each equivalence class and y-axis represents number of records in that equivalence class. Among these t-closeness values, we need to choose one by balancing data privacy v/s. utility of the data. The higher the threshold, greater the data loss and vice versa. Note that out of all the equivalence classes, we only selected those which are k-anonymized as a pre-requisite for this implementation.

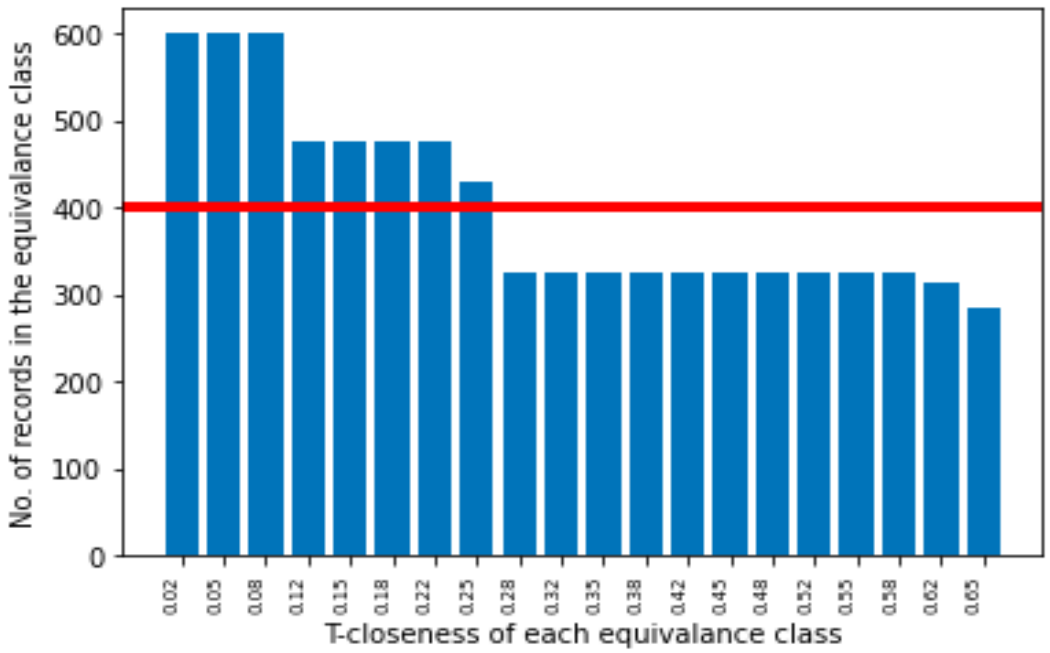

Figure 5.3: Data loss for each threshold value

For this implementation, the threshold value that is close to average data loss was

selected, which is t=0.16. In real world setting, the threshold value selection depends upon the data curator and how sensitive the data is. Now, lets visualize the records under each equivalence class which are above and below the threshold value we have selected. In each of the visualizations, the red line represents the value of t = 0.16, while x-axis represents the list of unique sensitive attributes and y-axis represents their t-closeness value in the respective equivalence class.

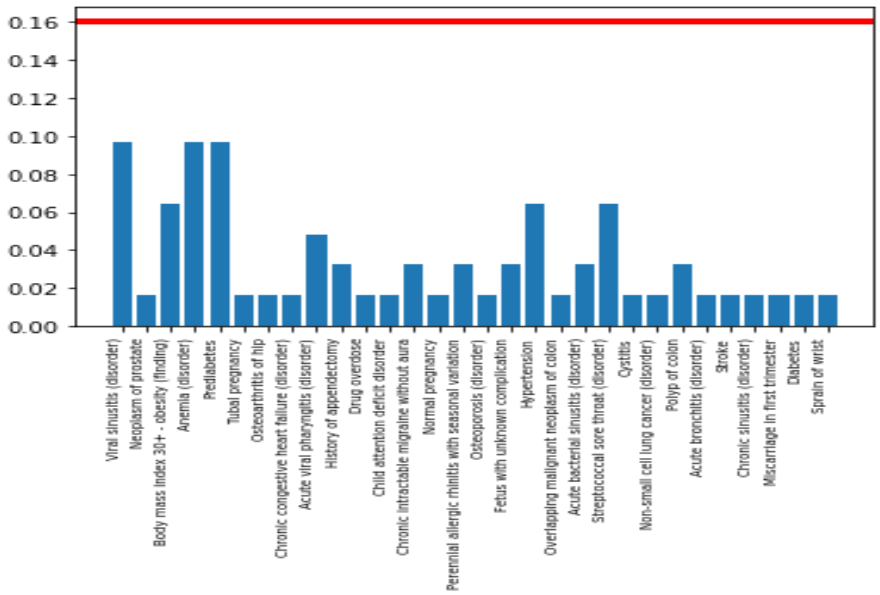

Figure 5.4: Equivalence class 2

In figure 5.4, which represents equivalence class 2, we see all the records are below the t-closeness threshold we selected, therefore we cannot include records from this equivalence class while publishing the data. Similar ideology is applied for the remaining visualizations.

In figure 5.5, for equivalence class 3, all the records except “Viral Sinusitis (disorder)” (records from fifth bar in the chart) and “Miscarriage in first trimester” (records from sixth bar in the chart) are below threshold. Therefore, we will only keep the records above the threshold. Similarly, in figure 5.6, we keep the records represented by the first, third and fifth bar charts for equivalence class 4. For equivalence class 5, only the records represented by second and third bar chart will be preserved since they are above our selected t-closeness threshold. In equivalence class 6, records representing both sensitive values are above threshold, therefore we include the entire equivalence class. Similarly, for equivalence class 7, 8 and 13, we see that all the

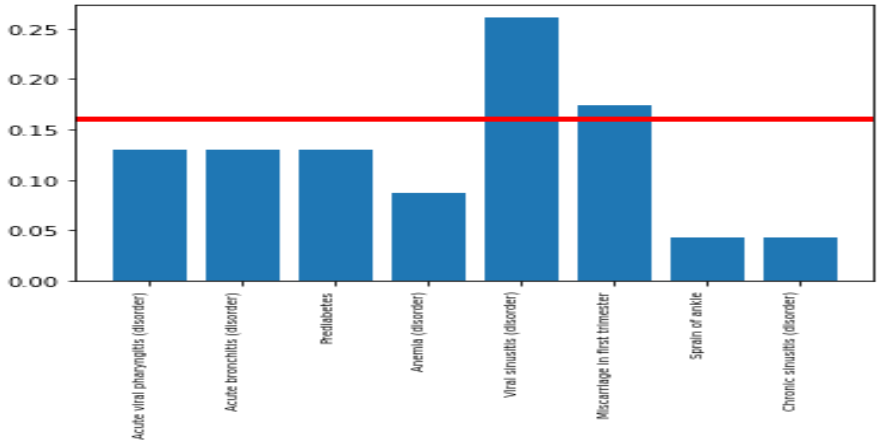

Figure 5.5: Equivalence class 3

records representing the respective sensitive values lie above the threshold and we consider the entire equivalence class respectively.

The one's below the threshold will be preserved and the others above the threshold will be suppressed from the final dataset according to the t-closeness notion. Note that we are only analyzing those equivalence classes which are k-anonymized which is a pre-requisite. Due to page size limitations, it was not possible to embed our data tables, pre-processing steps and our code, therefore we provide it in our replication package. The jupyter notebooks attached in the replication package validates our work and demonstrates how it works.

In this way, we could sanitize or anonymize the data before publishing it, giving us control over what we share and also data access monitoring could get more robust and reliable.

5.3 Differential Privacy

5.3.1 Introduction

The primary motivation of studying differential privacy came from the fact that differential privacy uses the notion of privacy budget which we will be discussing in section 5.3.7 and we find a potential here for our data usage monitoring tool which could be used to monitor privacy budget which is important factor in differential privacy. It is important to note that unlike l-diversity and t-closeness, differential privacy is a data aggregation technique and not a data anonymization tool. Differential pri-

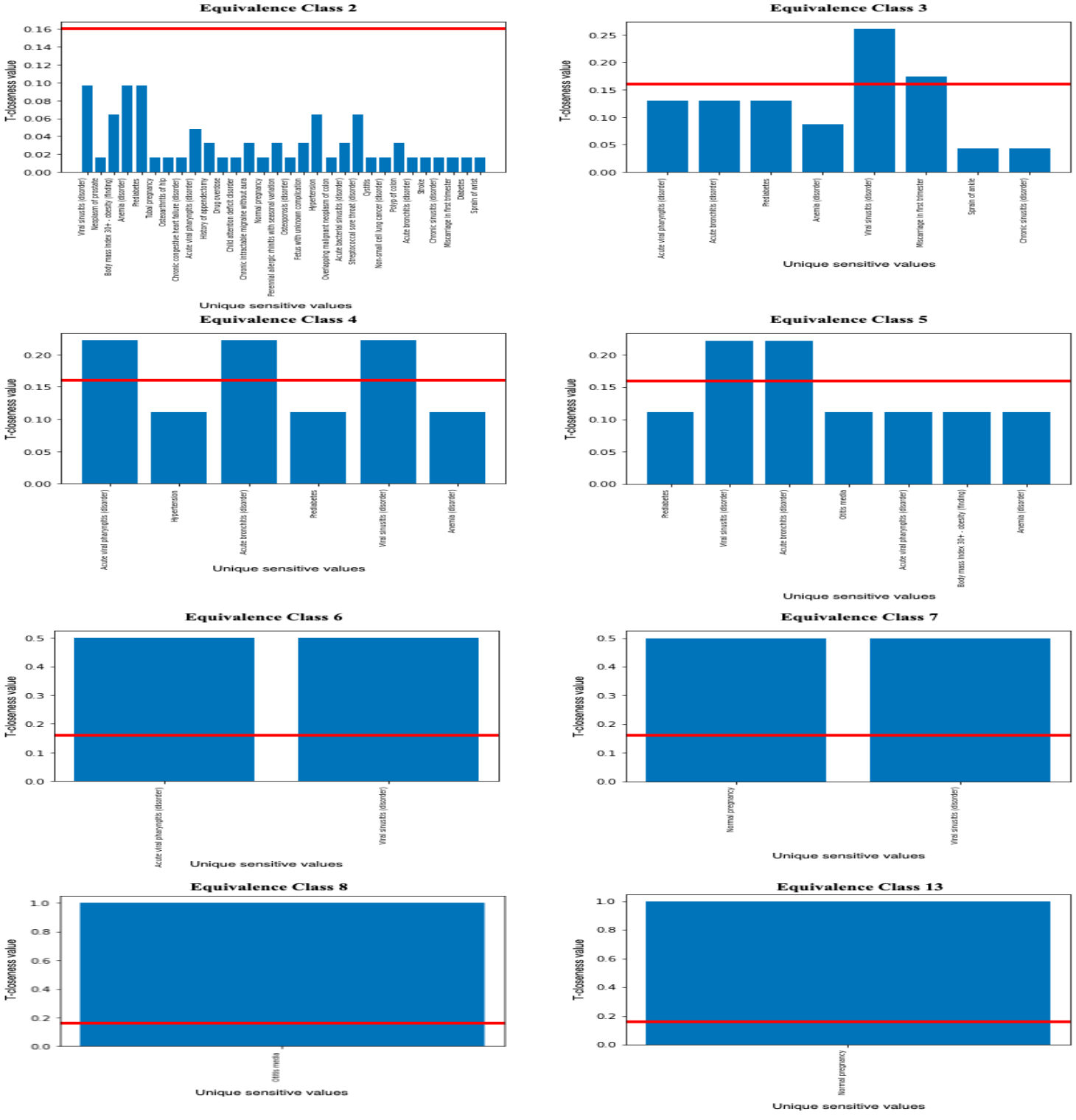

Figure 5.6: T-closeness of sensitive values within equivalence classes

vacy is a data privacy protection mechanism which comes by adding randomness or noise to the data before it is published or made available for analysis [13]. Which means, the more the noise the better the privacy. It is important to note the trade off between privacy of the data and the utility of the data itself. The threshold can be set according to the business context or specific requirements which is controlled using a privacy tuning parameter called epsilon (ϵ). The lower epsilon, higher the noise which means highest privacy but it also means less utility of the data. The lowest epsilon value provides the highest level of privacy but at the cost of the utility and the highest epsilon value provides the lowest privacy but high data utility which essentially might not be sufficiently differentially private.

Let us understand differential privacy with an example, let us assume that a child within a family does not like tomatoes in their diet. This information is personal to the child and their family. Although, the child goes to daycare along with three other children in total and the chef in the daycare cooks food for them, it important for the chef to know that some within the group of children does not want tomatoes in the diet and make the food accordingly for everyone, but it is key to note that the cook should not care which is the particular child that does not like tomatoes, rather should only know that there might be one or more children who does not like it in their diet. In this way, the private information is preserved yet the services are being improved without knowing the exact individual.

5.3.2 Mechanism of differential privacy

Name	Answer
Alice	Yes
Bob	No
Eliza	Yes
Frank	Yes
George	No
Mary	Yes
John	Yes

Table 5.1: Example dataset where 'Answer' is the sensitive attribute

Before moving into our implementation of differential privacy, we decided to discuss in brief about the mechanism of differential privacy. We got the inspiration of elaborating the mechanism of differential privacy came from [1], and the figures 5.7,

5.8 and table 5.1 is referred from the same source. Let us assume we have the above data source and the sensitive column being 'Answer' that we need to make differentially private, which means, adding noise to 'Answer'. The amount of noise depends upon the epsilon parameter as we learnt above and we also need some degree of randomness which can be statistically traceable. Let us understand it with the help of the following example.

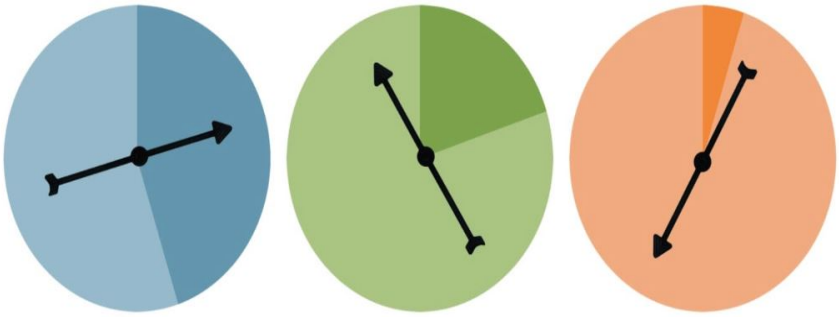

Figure 5.7: Spinners to represent the overview concept of differential privacy mechanism

The figure 5.12 illustrates three scenarios where each of them can be understood as having different epsilon values. The dial in the center is to have a degree of randomness. For the purpose of this example, let us define a simple noise addition mechanism: if the tip of the dial stops in the light area we do not change the 'Answer' and if it stops in dark area, we flip the 'Answer' i.e. if it is yes then change it to 'no' and vice versa. With this we have set up our noise addition mechanism. Also, we can correlate the light shade to our epsilon value; the lower the value, the higher the noise and privacy of data and vice versa. Let's say that our epsilon values are 0.55, 0.75 and 0.95 respectively for each of the above figures from left to right. After altering our results after each spin based on our noise addition mechanism, the randomized results are known to be differentially private. Let us say we have 100 records in a dataset and consider we spin the dials to get the below results:

Looking at the above results, we can see that the first figure where epsilon was 0.55, many spins landed in the dark area, whereas when it was 0.75 relatively less spins landed in the dark area and when it was 0.95 just 4 spins landed in the dark area. This directly helps us to understand the underlying concept of differential privacy in simple terms. In real world setting, Laplace [40] or exponential mechanism [40] is most popularly used for noise addition or randomness in epsilon differential privacy

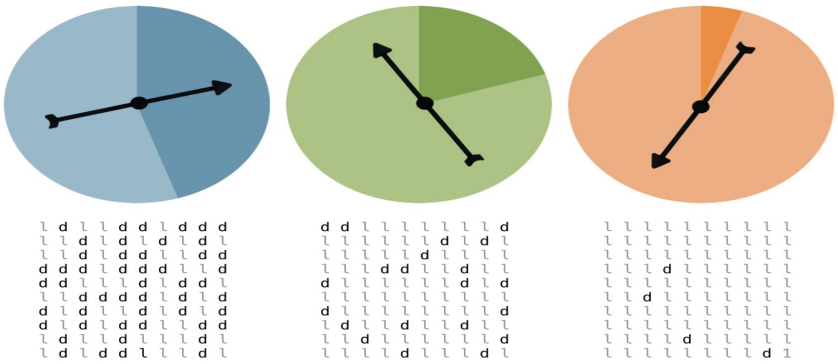

Figure 5.8: Output of each spinner after 100 spins

approach. But it is important to note that just adding noise or randomness does not ensure that data will be entirely private. This is because, if the attacker continuously issues the same query, even if there is noise in the data, it is possible to infer the patterns of randomness or noise and achieve a closer guess or exactly recognize the original information of the target. In order to tackle this, the concept of privacy budget was introduced.

5.3.3 Mathematical foundation

Sensitivity of dataset

In order to decide the privacy budget, it is good to know the sensitivity of the dataset to be made differentially private. In order to know the sensitivity, we apply a query function on two neighbouring subsets of a dataset [40]. We can make an assumption that these datasets differ from each other by one record. The sensitivity function gives us the maximum difference between the query results on the neighbouring datasets. This value helps determine the sensitivity of the dataset and make appropriate decisions in choosing a privacy budget [40]. In the equation below, let D be the collection of datasets, d be a positive integer, and $f : d- > R^d$ be a function. The sensitivity of a function denoted by delta(f), is defined by:

$$delta(f) = max||f(D) - f(D')|| \tag{5.1}$$

Laplace noise addition mechanism

The key mathematical component in the differential privacy is a mechanism to add noise. In this implementation, we used the Laplace Noise Addition mechanism. The Laplace distribution is similar to the Gaussian/normal distribution, but is sharper at the peak and has fatter tails. It represents the difference between two independent, identically distributed exponential random variables [1]. The first law of Laplace, from 1774, states that the frequency of an error can be expressed as an exponential function of the absolute magnitude of the error, which leads to the Laplace distribution. For many problems in economics and health sciences, this distribution seems to model the data better than the standard Gaussian distribution. It uses the following probability density function, where μ is a location parameter and $\lambda > 0$, which is often referred to as the diversity, is a scale parameter:

$$f(x; \mu, \lambda) = 1/2\lambda * exp(-\frac{|x - \mu|}{\lambda}) \tag{5.2}$$

The numpy function used in the project is:

```
np.random.laplace(0, 1.0/epsilon, 1)
```

Figure 5.9: Laplace noise addition using Numpy

Where the first parameter is the position of the distribution peak, which is 0 by default, the second parameter is the exponential decay which is a key parameter to compute the noise to be added which is 1/epsilon in our case. The numerator is the sensitivity value, and it is one in our case because the project only considers count functions and the sensitivity of count function is 1.

5.3.4 Privacy budget composition

We believe our data access monitoring tool has a potential to play an important role in monitoring the privacy budget. To control the number of queries issued to the dataset in order to prevent inference from randomized results, the concept of privacy budget comes in place. The lower the epsilon the higher the privacy budget and vice versa as we learnt in section 5.3.2. If we think of the privacy budget in terms of money,

then let us assume we have a budget of 1 CAD [4]. This budget can be exhausted by issuing a single query that costs 1 CAD (provides all/enough information to the issuer by preserving privacy, if any further query is issued the privacy can be compromised) or one query that costs 0.60 CAD and 4 queries costing 0.10 CAD can be issued. With this we ensure that all the queries issued to the data comply within the privacy budget for each individual giving less opportunity to the issuer/attacker to infer the original data from the randomness of the data. Although, there is one challenge with this approach. If the query issuer issues ten 0.10 CAD queries, it is still possible to get different results for each of those 10 queries which could be sufficient enough to infer the originality from the randomness whereas everything is within the privacy budget. This problem can probably be addressed by caching the queries for each individual so that if the same query is issued, the same result is generated and also the privacy budget is not exhausted for the repetitive queries.

5.3.5 Our implementation

The implementation makes use of the architecture which we explained in Figure 5.15. The requirements of our implementation were generated by understanding a specific scenario as stated as below:

- A data curator owns important anonymous statistical information about the patients (Synthea database [3]) within the healthcare system which is also sensitive in nature.
- It also becomes necessary that this data should be shared to the group of people or an individual who may make use of this data to help improve the healthcare system to serve the patients better or make similar contributions.
- At the same time, the data curator needs to make sure to protect the data against the data privacy related attacks, the most popular being linkage attacks [22].
- The data curator decides to implement a system that provides the data consumers with enough information for carrying out statistical analysis successfully and yet not able to link or know who the particular individuals are in the dataset.

5.3.6 System architecture and specifications

System Specifications:

- *Mechanism:* Pure differential privacy mechanism (Epsilon Differential Privacy).
- *The noise inducing mechanism:* Laplace.
- *Privacy budget composition:* Parallel.
- *Query function support:* Count.

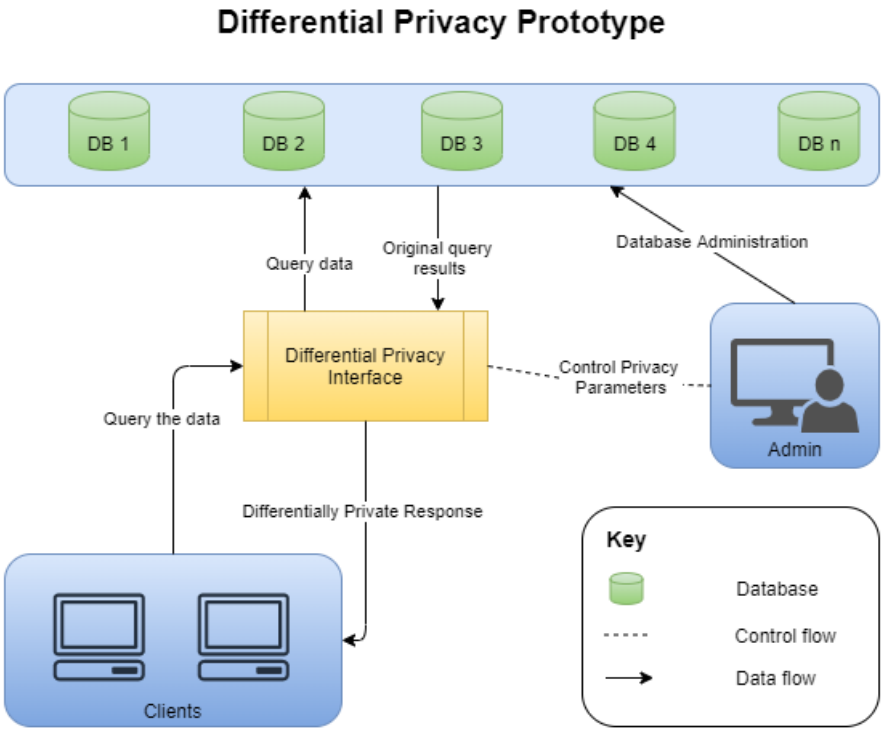

Figure 5.10: Differential privacy prototype

It is important to note that in order to maintain simplicity, ℓ-diversity and t-closeness are not included in this implementation although they could be used as pre-requisites to the system. In the system architecture shown in figure 5.15, clients are the entities which request for the statistical data to the data curator (Admin). In the project scope, the clients can currently only issue count queries. The Admin manages the control of the differential privacy system and also has the ability to administer the database. The control parameters in differential privacy include assigning epsilon

value to each data set for balancing the noise vs. utility. Admin can also be responsible to add new users to the system and process or deny any special requests from the users if there are any. There are two views of the project:

- Admin (curator) View
- Client (user) View

A. Admin (curator) View:

This view is meant for the data curator to control the differentially private framework. The curator is a trusted entity who has access to all the raw assets (databases/tables). Depending on the sensitivity of the dataset, the curator can adjust the value of epsilon for each asset which is responsible to control the amount of noise being added to the counts. Besides, the curator also has a view of the count of users using the system, total query transactions that took place and the epsilon for each asset. Also, the curator can view the users who exhausted their privacy budget. Finally, the curator can also decide which attributes of the asset are visible to the users and which are to be kept hidden.

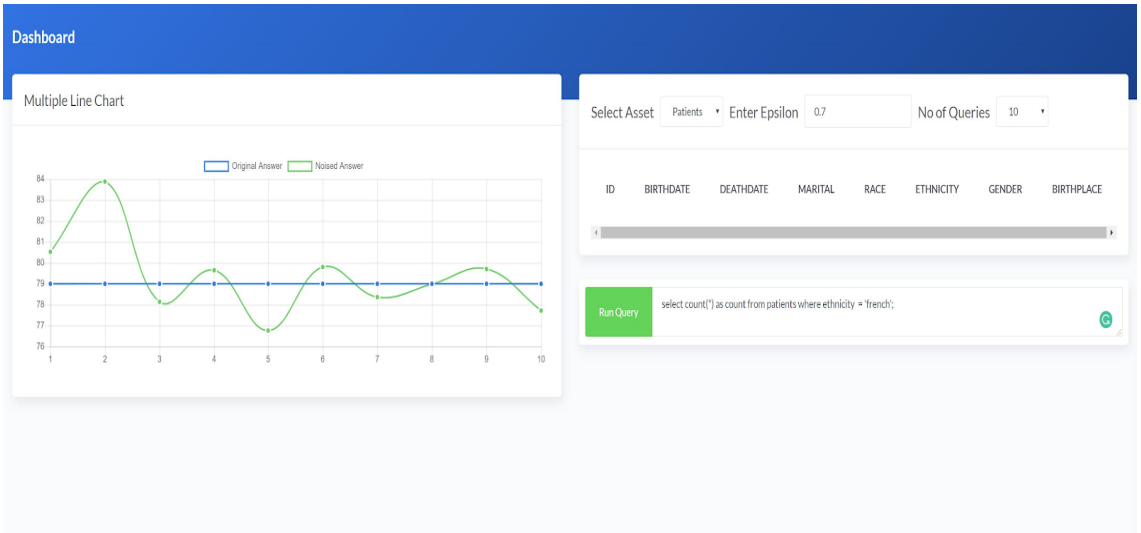

Figure 5.11: Admin epsilon selection and query results after 10 queries

Figure 5.17 indicates the ability of admin to determine optimal epsilon value for a particular data asset. The admin is able to view a visualization of utility vs. noise in the data and can make appropriate decisions on the optimal value of epsilon based on the sensitive nature of the dataset. The blue line in the chart indicates the original

answer and the green line indicates noised response. The x-axis is the number of queries while the y-axis is the query response.

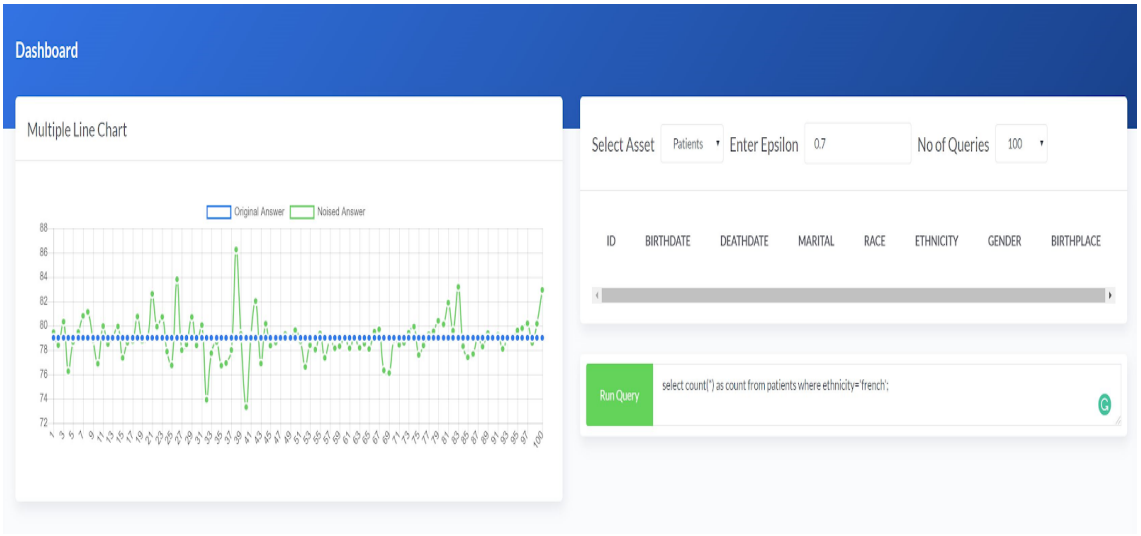

Figure 5.12: Admin epsilon selection and query results after 100 queries

Similarly, figure 5.18 demonstrates the visualization when after running 100 queries. The key point to note is that the epsilon value for these visualizations is 0.7, but the chart changes when the epsilon values change as it is the parameter to tune the amount of noise being added.

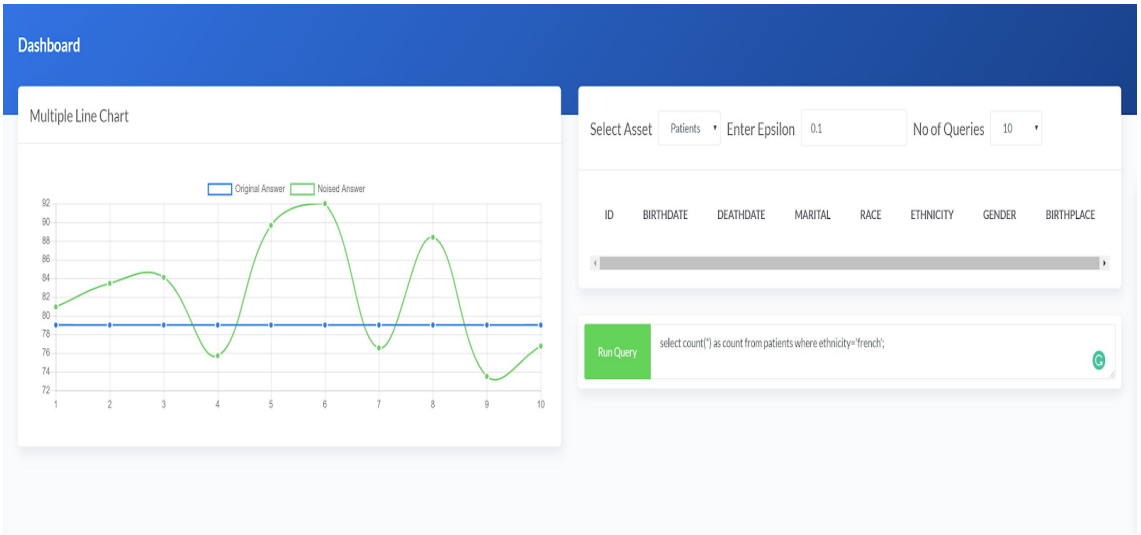

Figure 5.13: Admin epsilon selection as 0.1 (highest noise) and query results after 10 queries

Figure 5.19 demonstrates the visualization when the epsilon value is 0.1, which is

the highest noise in the data and how it can affect the utility since there is a significant difference between the original answers and noised answers.

B. Client (user) view:

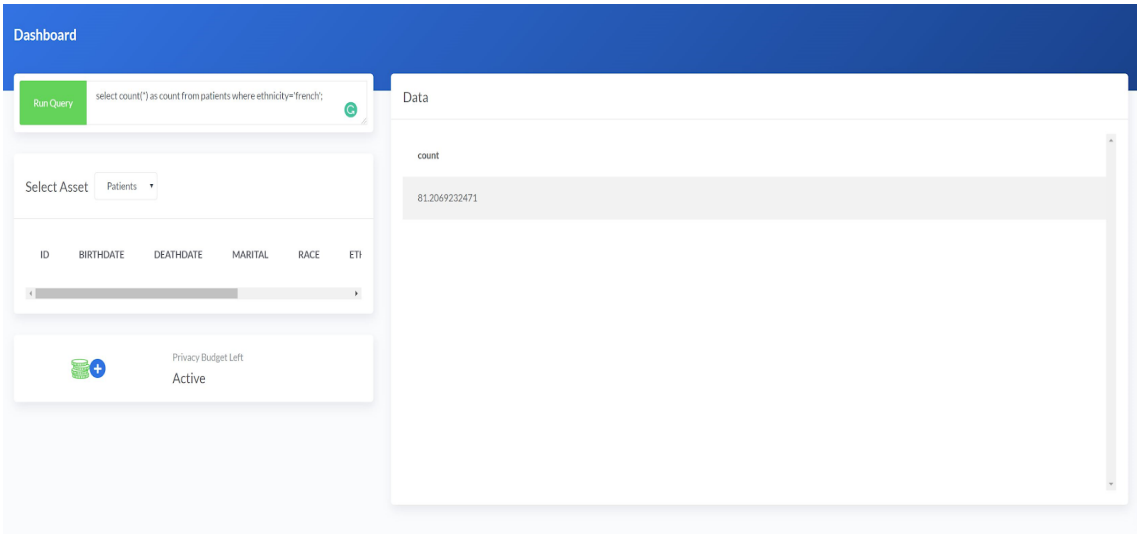

Figure 5.14: Count query execution by the user, privacy budget not yet exhausted

The user view is for the data requestor to access the statistical information in a differentially private manner. The user can select an asset from the list which is allocated by the data curator. After selecting the asset, the user gets a view of all the attributes allowed for the user to query. The user then runs a 'count' query on the given set of attributes. (Currently, the system supports only 'count' query functions). Then the user receives the differentially private results with noise added to the query results. But there is a limit to the number of queries the user can issue which is kept hidden from the user and is decided at the runtime. This is to address the limitation of differential privacy, the limitation being that after certain amount of queries, the pattern of the randomness of noise can be identified, therefore, the system anticipates at runtime when the query issuer might get to know about the random pattern and disables the query functionality for the user.

In this experiment, the user is not aware of the privacy budget allocation in advance, rather the system analyzes the privacy budget at run-time and when the user is close to getting a real answer (we get to know by computing mean after every count query output) the privacy budget is automatically exhausted for the user and

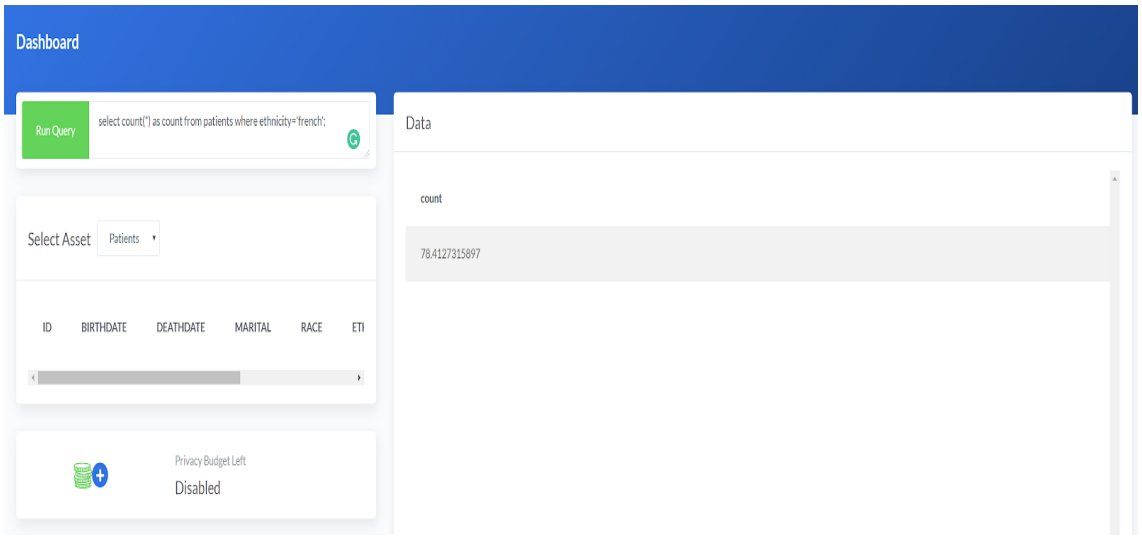

Figure 5.15: The privacy budget exhausted by the user and is now disabled to query on that asset

Result of count query	Original count	Average of results	Privacy budget
80.027	79	80.027	Active
81.970	79	80.998	Active
73.063	79	78.353	Active
82.01	79	79.267	Active
78.412	79	79.096	Disabled

Table 5.2: Disabling privacy budget when average of results is close to the original answer by +/- 0.01

the user can no longer query that particular asset, i.e. if the mean of the result of the count query is closer to +/- 0.01 to the original output, then the privacy budget is exhausted. The threshold of 0.01 was selected randomly, although data curator can decide based on sensitivity of the dataset. Table 5.2 validates the output of our differential privacy prototype. After each query output, the average of results is calculated, and soon as the average reaches +/- 0.01 of the original output which is 79, the privacy budget is disabled. Figure 5.14 illustrates the idea of disabling the privacy budget for the user when the user reached closer to the exact answer thereby disabling the user to issue queries to the system. This is the key component of a differential private system. As the system is designed to add random noise to the query results, the random results can get predictable after issuing a certain amount of queries to the dataset which might reveal the correct answers to the user. To prevent

this, the concept of privacy budget carries a lot of importance in a differential private system.

5.4 Limitations

We only studied 3 popular techniques to implement data privacy, although, there could be many other techniques in the literature which we did not study during the course of this research. Also, the differential privacy tool currently only supports the count queries since calculating sensitivity for the same is straightforward. There is still a scope to support other types of queries including joins and aggregations which was not a part of this study. Additionally, we did not build a robust user role and authentication system, therefore, presently only admin approved users can get access to the system which could be thoroughly verified before allowing them access to prevent creation of multiple accounts by the same user. The differential privacy implementation attempted to implement the theoretical knowledge gained about differential privacy from the available literature into a real-world prototype. Although the concept of differential privacy is well defined theoretically, some of the key concepts were hard to implement practically. For example, the literature conveys the idea of privacy budget and how it plays an important role in the entire framework of differential privacy, but little is available to understand this concept in general terms and how it could be applied to any dataset which we understand from studying the relevant research papers in our references. Also, differential privacy cannot be a complete solution for offering data privacy if we also need data anonymization protocols. Differential privacy helps to prevent knowing the exact sensitive details by inducing noise, but does not offer anonymization solutions like k-anonymity, l-diversity or t-closeness within itself, all these techniques could be orchestrated together in different requirement scenarios.

5.5 Chapter summary

In this chapter, we attempted to find answer to our RQ 3: “Which techniques could be used to secure sensitive information before data publishing to gain better control over data sharing?”. In chapter 3, we discussed about the specifics of our tool for data usage monitoring and in chapter 4 we found that this tool could be applicable in broader industry segments especially under the audits and access control tactic,

specifically to monitor the usage of data. In addition to data usage monitoring, we also proposed the idea of securing the data before it is shared for controlled data distribution which would make the end-to-end system robust. Therefore, we introduced an extension to our data usage monitoring tool architecture that is aimed to enable data anonymization and controlled data sharing through the techniques we studied in this chapter. Depending on the data sharing requirements, all the techniques we discussed in this chapter could be used together in orchestration or only some of them could be used depending on different requirement scenarios. ℓ-diversity and t-closeness ensure that the data is sufficiently sanitized or anonymized based on the rules set by the data curator before sharing the data, whereas, differential privacy helps us to control the access to this shared data by introducing the notion of privacy budget and limiting the exposure of the accurate sensitive data by introducing noise in shared data. Our tool for data usage monitoring adds an additional layer of vigilance over monitoring the queries issued to the shared data and reporting any unusual query that violated data sharing agreement to the data curator. Additionally, in the future, our tool could also help in monitoring privacy budget for every consumer of the data in order to make differential privacy component in this architecture more robust.

Chapter 6

Discussion

In this chapter we discuss our system architecture as a whole, research and practical implications of our work and reflect upon the findings of our research questions. From the development of our data usage monitoring tool, to conducting an industrial survey and to finding out the applicability of our tool in broader industry sector then designing a secure data publishing extension, we were able to implement an end to end secure data publishing and data usage monitoring tool for the use case of our research partner. The figure 6.1 highlights the system architecture of our tool. Let's discuss each of these components:

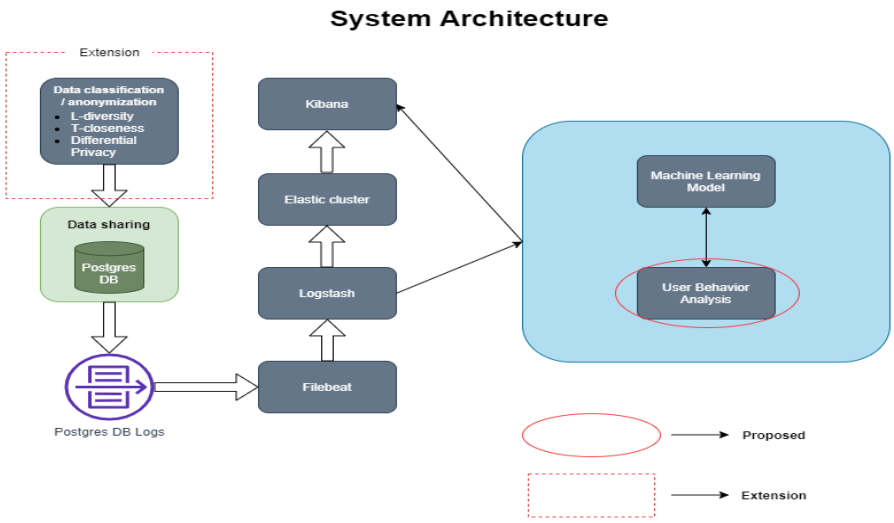

Figure 6.1: System architecture - replicated from section 3.4

1. Data classification / anonymization: This module is an extension to our existing architecture for data usage monitoring tool which we discussed in detail in

chapter 5. Currently, the data curator can choose between ℓ-diversity, t-closeness or differential privacy mechanisms to anonymize or sanitize the private data. This is a key module that helps anonymize the data and quantify data privacy before the data is being published, making our system an end-to-end secure data publishing and data usage monitoring platform.

2. Data Sharing: This phase is responsible to share the private and anonymized data among the data consumers. The data sharing can be done using data sharing agreements or blockchain and data could persist in any database server, currently we support Postgres SQL database.

3. Filebeat: This component is a data shipper for logs. It continously watches for new logs in the system and it will directly bring the logs from Postgres database log directory and supply them to Logstash after every specified time intervals.

4. Logstash: It does the job of log data processing and transformation of data as required by the ML model and Elastic cluster.

5. Machine Learning Model: This component holds the logic for predicting anomalies and detecting suspicious patterns which is trained using one-class SVM clustering approach.

6. Elastic-cluster: The search engine that indexes the transformed data used for searching everything (anomalies, statistics etc.) about the log data.

7. Kibana: Our data visualization dashboard for visualizing insights.

8. User Behavior Analysis: This component is expected to be a separate machine learning model that analyzes user behaviour or access patterns through logs. We propose this as the future scope for this research project and this is why it is highlighted in the red oval.

In chapter 2 we discussed the background of our use case and the requirements for which we began developing this tool. Initially, our tool was limited only for the data usage monitoring purpose. Later, we decided to conduct an industrial survey to know about the tactics followed by the data driven developers to help maintain data privacy at work and we learnt that majority of our participants were at initial stage of incorporating data privacy in their work. Furthermore, we found out that 'Audits and access control' is the most commonly discussed tactic among the data driven developers to monitor data usage and maintain data privacy. Through this we realized the scope of our data usage monitoring tool in the broader industry sector. This motivated us further to develop an end-to-end data publishing and data usage

monitoring tool by adding support to use secure data publishing techniques. We studied the three most popular techniques in data privacy - ℓ-diversity, t-closeness and differential privacy - to support our idea of secure data publishing by demonstrating the implementation of each of these techniques in chapter 5. In the following sections, we will be discussing the implications of our research work in both research scope as well as practical scope.

6.1 Audits and access control is the preferred tactic to monitor data access within organizations

One of the most common approach during the initial stage of incorporating data privacy regulations, understanding from our survey responses, is to have access control mechanisms in place for data access and monitoring the data usage. The data usage monitoring tool which we developed and discussed in chapter 2, attempts to automate this process and make it reliable to identify any potential data regulation or data agreement violations. Understanding from the responses of our participants, it was known that most of them were at the preliminary stage of adopting data privacy regulations and audits and access control is the most preferred tactic among them to monitor data usage while they figure out the way to understand and comply with data privacy regulations.

Implications

Data usage monitoring is the most common approach to ensure ethical use of data by the consumers, may those be developers or clients. However, the data usage policies and data agreements are established using natural language text and mapping them into programming rules is still a challenge, especially for complex privacy policies or data agreements. Therefore, more research is needed to efficiently map the natural language documents to programmable rules so that they could be easily fed into our data usage monitoring tool. The use of smart contracts using blockchain is also another area of research that could make this task achievable.

Research implication 1

How could we efficiently map the privacy policies or data sharing agreements in natural language into programmable rules for reliable and robust data usage monitoring and access control?

For effective transfer of knowledge from natural language text to the programmable rules, it is important that each developer is adequately aware of the data privacy document of their organization. The privacy regulations and processes within the organizations also must be transparent to the developers to gain breadth of understanding on data privacy requirements within the organization.

Practical implication 1

Every developer must be aware of the data privacy regulations within their organization and the privacy education regarding privacy policies and regulations must be offered to the developers by their organization.

6.2 Understanding and implementing the data privacy regulations at work is still a challenge

We conducted a literature review of the existing survey based studies among the developers on their perception of data privacy. Even though, the studies were conducted among general developers, it is a possibility that the participants may or may not be data driven / machine learning developers. We identified this gap and decided to conduct a survey study particularly among the data driven developers within the industry. We tried to bridge the knowledge gained from the literature review and our goals. Our survey was specifically targeted to the decision makers and data driven developers, and we learnt that our participants were aware of the data privacy regulations, however, most of the organizations were at an initial stage of incorporating data privacy in their development process and therefore the adoption is not yet widespread. Some of the participants that were decision makers in mid-size and large organizations indicated that they find it challenging to understand the data privacy regulations and try to hire a third party for conducting frequent audits.

Implications

The data privacy requirements might differ for every organization and therefore there is a need to understand the general data privacy regulations and engineer them into the organization's development workflow, which for now looks like a manual process. More research is needed in the direction of making the understanding of data privacy regulations less challenging to developers and generalizing the design practices and tactics based on different use cases also needs to be documented.

Research implication 2

How to generalize and make it easier and faster to incorporate data privacy regulations in the development workflow?

Apart from making the existing data privacy rules less challenging to understand, some steps can also be taken within the organization, such as conducting training among across teams and aligning the data privacy regulations with the requirements of a particular project or a team. This might make incorporating data privacy regulations in the work less challenging.

Practical implication 2

Frequent and role based training on understanding data privacy regulations and how they align with organizational requirements should be conducted within each team that deal with private data.

6.3 Anonymizing data and control over data sharing helps in promoting secure data sharing environment

The use of data usage monitoring will help in automating and identifying unusual data access requests, although, having an additional layer of securing data even before sharing it makes the entire data sharing and usage monitoring pipeline more reliable and robust. Out of the many data anonymization and controlled data sharing techniques, we implemented ℓ-diversity, t-closeness and differential privacy to

demonstrate how these techniques could help in secure data publishing. It is again a part of the audits and access control tactic which gives helps us gain a degree of access control over the data before publishing it by anonymizing it using ℓ-diversity and t-closeness or adding noise to the data using differential privacy.

Implications

Further research could be carried out in terms of exploring other techniques and implementing them into our tool, this would allow the data curator to choose among multiple techniques and orchestrate them together based on different data publishing requirements.

Research implication 3

Which other secure data sharing techniques need to be incorporated into existing tool to enable support for different secure data publishing requirements?

Secure data publishing requirements might differ from projects to projects, but the underlying approach to address the problem may not change drastically.

Practical implication 3

In the practical working environment, the definition of sensitive data may change from projects to projects, for example, access to the name of the disease of a patient may be sensitive information for one team but not for the other who is using this data to derive insights.

6.4 Findings

This section highlights the research questions that drove this thesis and connects them to the findings that answer them.

RQ 1: How machine learning can be used as a "Audits and access control" tactic to maintain the quality of data privacy?

Finding 1: With the data usage monitoring tool that we developed for the use

case of our research partner, we realized that machine learning can play an important role in automating the data usage monitoring task and can make the process much reliable. After studying several methods for monitoring data access control using machine learning, we learnt that log monitoring and anomaly detection is the reliable way to achieve it. We studied methods of data collection, log processing, rule based control and use of machine learning for log monitoring. After gaining an understanding of the foundations, we realized identified the potential of machine learning to solve the problem of data usage monitoring for our use case which we discussed in chapter 2. We decided to build a simple yet generic tool which would address the issue of monitoring data access of the data consumers which may include a formal data sharing agreement between the parties. In this way, we attempted to demonstrate how machine learning can be helpful in maintaining quality of data privacy through data access monitoring and serve as a key component for "Audits and access control" tactic to maintain vigilance over data sharing among data driven developers, which we found out through our industrial survey.

RQ 2: Which tactics do the data driven practitioners in the industry follow or suggest to ensure data privacy?

Finding 2: After developing the data usage monitoring tool for our use case described in chapter 2, we decided to find out the applicability of our tool in the broader industry sector through a survey. The industrial survey was targeted towards key decision makers and data driven developers in within the industry. Using the design science approach, we found out that our tool has a scope of application within the industry. Majority of the responses of our participants were in the category of audits and access control followed by data related operations. After thematic analysis and categorization of responses, we found 4 key tactics practiced by the data driven developers in the industry, which are 'Audits and access control', 'Data related operations', 'Privacy awareness' and 'Machine learning protocol'. Although we acknowledge the low number of participants in our study, majority of the participants were decision makers in their organization which added quality into the findings of our study. Based on the analysis of their responses, we can learn that 'Audits and access control' was the most discussed tactic among the industrial participants thereby indicating a scope of applicability of our data access monitoring tool to address this problem.

RQ 3: Which techniques could be used to secure sensitive information before data publishing to gain better control over data sharing?

Finding 3: Currently we studied 3 data privacy techniques during the course of this research. Two of them (ℓ-diversity and t-closeness) being data anonymization methods and the third (differential privacy) being the controlled data publishing mechanism. The motivation to study these techniques came from our need to extend our existing data usage monitoring tool with secure data publishing mechanism to implement an end-to-end application from secure data publishing to monitoring of data usage. We further understood how these techniques could be used together along with our existing data usage monitoring tool and make the end-to-end data publishing and usage monitoring process more secure and controlled.

Chapter 7

Conclusion

In many organizations, privacy policies and their broader privacy climate are not always aligned. Understanding the tactics the data driven developers use within their organizational climate helped us to know their perception towards data privacy and actions they take to ensure the same. We started with the funded project where we built a tool to monitor data access license violations or data usage monitoring. This motivated us to find out the applicability of this tool in the industry which led to carry out a design science approach by conducting an industrial survey to understand what are the data privacy challenges faced in the industry and what tactics do they follow. After conducting an industrial survey, we learnt that access control is still the most preferred tactic to address and maintain data privacy. We realized that the system architecture which we developed during applies to a broader industry sector which attempts to provide a solution for monitoring data access and detecting anomalous data access attempts. We also extended our existing architecture with a secure data publishing module which could also be used for measuring data privacy into quantifiable terms which enables control over data publishing. This led us to further conduct research in this direction. We explored 3 techniques, namely, l-diversity, t-closeness and differential privacy. We also attempted to show how we could use these techniques in practice to achieve secure data publishing and have a better control over secure data sharing. Although, it is important to note that none of the approach within the architecture is self sufficient by itself to detect or prevent violations of data sharing agreement. As we learnt, rule based access control does play a pro-active role in preventing data sharing agreement violations but it's limitations are covered by the machine learning model as we learnt in chapter 2. Additionally, the machine learning model also had limitation of ignoring false negatives which could

be balanced by predictive user access pattern analysis. Therefore, multiple similar components would be needed to balance the limitations of each other. Finally, we believe that our proposed system architecture would be a foundation especially for end-to-end secure data sharing and data usage monitoring.

Future work

This section highlights the areas where the work demonstrated in this thesis can be taken further.

User behavior analysis through logs

Currently, we trained our machine learning model based on rules specified in the data sharing agreement. Although, there is possibility to identify the flaws in the pre-defined rules and could be used in favor of violation of data sharing agreement which may not be detected using our current machine learning model. Studying the behavior of the user through data access logs will enable us to know in advance if the violation is going to take place which will make the system pro-active rather than notifying after violation takes place.

Machine learning to detect PII

The work throughout the thesis focused entirely on the data access control monitoring and secure data publishing. Although, there was a second most discussed tactic which we found among the participants in our survey which was "Data related operations". Through the responses it was indicated that PII can sometimes become a very sensitive part of the data and therefore having a machine learning model to detect PII instead of manual detection could help add a lot of value into data driven developer community.

Validation of the machine learning model

We trained our machine learning model using One class SVM approach with a minimal dataset available to us. Also the dataset was synthetically generated therefore there is a high possibility of degree of bias while training the model. Additionally, the model could be trained using other approaches and validated to make the predictions more reliable.

Exploring additional techniques for secure data sharing
In our extension module, we studied the three stated techniques for secure data publishing since these were the most popular. Although, there is scope to implement or orchestrate together additional techniques from the literature.

Extending differential privacy implementation to support additional queries
Currently, our differential privacy method only supports the count query as the sensitivity function was easy to determine. For other types of queries, calculating the sensitivity could get complex. Extending this work to support additional types of queries would improve the applicability spectrum of the existing differential privacy tool.

Bibliography

[1] https://accuracyandprivacy.substack.com/.

[2] https://libraryguides.mcgill.ca/c.php?g=478294p=3270938.

[3] https://synthetichealth.github.io/synthea/.

[4] https://www.johndcook.com/blog/2019/10/14/privacy-budget/.

[5] https://www.tf-pm.org/resources/casestudy/neste-big-data-is-the-new-oil.pdf.

[6] Oshrat Ayalon, Eran Toch, Irit Hadar, and Michael Birnhack. How Developers Make Design Decisions about Users' Privacy: The Place of Professional Communities and Organizational Climate. In *Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing - CSCW '17 Companion*, pages 135–138, Portland, Oregon, USA, 2017. ACM Press.

[7] Andy Brown, Aaron Tuor, Brian Hutchinson, and Nicole Nichols. Recurrent Neural Network Attention Mechanisms for Interpretable System Log Anomaly Detection. *CoRR*, abs/1803.04967, 2018.

[8] Q. Cao, Y. Qiao, and Z. Lyu. Machine learning to detect anomalies in web log analysis. In *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, pages 519–523, December 2017.

[9] Ann Cavoukian, Scott Taylor, and Martin E. Abrams. Privacy by Design: essential for organizational accountability and strong business practices. *Identity in the Information Society*, 3(2):405–413, August 2010.

[10] Josep Domingo-Ferrer, Marit Hansen, Jaap-Henk Hoepman, Daniel Le MÃľtayer, Rodica Tirtea, Stefan Schiffner, George Danezis, European Union, and European Network and Information Security Agency. *Privacy and data protection by design - from policy to engineering.* ENISA, Heraklion, 2014. OCLC: 903502987.

[11] Min Du, Feifei Li, Guineng Zheng, and Vivek Srikumar. DeepLog. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security - CCS \textquotesingle17*. ACM Press, 2017.

[12] Sizhong Du and J. Cao. Behavioral anomaly detection approach based on log monitoring. In *2015 International Conference on Behavioral, Economic and Socio-cultural Computing (BESC)*, pages 188–194, October 2015.

[13] Cynthia Dwork. Differential Privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*, pages 1–12, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[14] Khaled El Emam and Fida Kamal Dankar. Protecting Privacy Using k-Anonymity. *Journal of the American Medical Informatics Association*, 15(5):627–637, September 2008.

[15] J. Glasser and B. Lindauer. Bridging the Gap: A Pragmatic Approach to Generating Insider Threat Data. In *2013 IEEE Security and Privacy Workshops*, pages 98–104, May 2013.

[16] Mark A. Griffin and Andrew Neal. Perceptions of safety at work: A framework for linking safety climate to safety performance, knowledge, and motivation. *Journal of Occupational Health Psychology*, 5(3):347–358, 2000.

[17] Irit Hadar, Tomer Hasson, Oshrat Ayalon, Eran Toch, Michael Birnhack, Sofia Sherman, and Arod Balissa. Privacy by designers: software developers' privacy mindset. *Empirical Software Engineering*, 23(1):259–289, February 2018.

[18] Ruogu Kang, Laura Dabbish, Nathaniel Fruchter, and Sara Kiesler. âĂIJMy Data Just Goes Everywhere:âĂİ User Mental Models of the Internet and Implications for Privacy and Security. page 14.

[19] Ninghui Li, Tiancheng Li, Suresh Venkatasubramanian, and T Labs. t-Closeness: Privacy Beyond k-Anonymity and -Diversity. page 10.

[20] Eduardo Lopez and Kamran Sartipi. Feature Engineering in Big Data for Detection of Information Systems Misuse. In *Proceedings of the 28th Annual International Conference on Computer Science and Software Engineering*, CASCON '18, pages 145–156, Riverton, NJ, USA, 2018. IBM Corp. event-place: Markham, Ontario, Canada.

[21] S. Lu, B. Rao, X. Wei, B. Tak, L. Wang, and L. Wang. Log-based Abnormal Task Detection and Root Cause Analysis for Spark. In *2017 IEEE International Conference on Web Services (ICWS)*, pages 389–396, June 2017.

[22] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. âĎŞ-Diversity: Privacy Beyond k-Anonymity. page 12.

[23] E. Marchi, F. Vesperini, F. Weninger, F. Eyben, S. Squartini, and B. Schuller. Non-linear prediction with LSTM recurrent neural networks for acoustic novelty detection. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, July 2015.

[24] Rafael P. MartÃŋnez-ÃĄlvarez, Carlos Giraldo-RodrÃŋguez, and David Chaves-DiÃľguez. Large scale anomaly detection in data center logs and metrics. In *Proceedings of the 12th European Conference on Software Architecture Companion Proceedings - ECSA \textquotesingle18*. ACM Press, 2018.

[25] Stan Matwin, Jordi Nin, Morvarid Sehatkar, and Tomasz Szapiro. A Review of Attribute Disclosure Control. In Guillermo Navarro-Arribas and VicenÃğ Torra, editors, *Advanced Research in Data Privacy*, pages 41–61. Springer International Publishing, Cham, 2015.

[26] Marie Caroline Oetzel and Sarah Spiekermann. A systematic methodology for privacy impact assessments: a design science approach. *European Journal of Information Systems*, 23(2):126–150, March 2014.

[27] K. Sudheer Reddy, M. Kantha Reddy, and V. Sitaramulu. An effective data preprocessing method for Web Usage Mining. In *2013 International Conference on Information Communication and Embedded Systems (ICICES)*, pages 7–10, February 2013.

[28] J. Roh, S. Lee, and S. Kim. Anomaly detection of access patterns in database. In *2015 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1112–1115, October 2015.

[29] M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440, Dec 2012.

[30] Awanthika Senarath and Nalin A. G. Arachchilage. Why developers cannot embed privacy into software systems?: An empirical investigation. In *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018 - EASE'18*, pages 211–216, Christchurch, New Zealand, 2018. ACM Press.

[31] Awanthika Senarath, Marthie Grobler, and Nalin Arachchilage. A Model for System Developers to Measure the Privacy Risk of Data. 2019.

[32] Awanthika Rasanjalee Senarath and Nalin Asanka Gamagedara Arachchilage. Understanding Organizational Approach towards End User Privacy. page 12, 2017.

[33] Swapneel Sheth, Gail Kaiser, and Walid Maalej. Us and them: a study of privacy requirements across north america, asia, and europe. In *Proceedings of the 36th International Conference on Software Engineering - ICSE 2014*, pages 859–870, Hyderabad, India, 2014. ACM Press.

[34] Latanya Sweeney. k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, October 2002.

[35] Uttam Thakore, Ahmed Fawaz, and William H. Sanders. Detecting monitor compromise using evidential reasoning. In *Proceedings of the 5th Annual Symposium and Bootcamp on Hot Topics in the Science of Security - HoTSoS \textquotesingle18*. ACM Press, 2018.

[36] Aaron Tuor, Samuel Kaplan, Brian Hutchinson, Nicole Nichols, and Sean Robinson. Deep Learning for Unsupervised Insider Threat Detection in Structured Cybersecurity Data Streams. *CoRR*, abs/1710.00811, 2017.

[37] M. Wang, L. Xu, and L. Guo. Anomaly Detection of System Logs Based on Natural Language Processing and Deep Learning. In *2018 4th International Conference on Frontiers of Signal Processing (ICFSP)*, pages 140–144, September 2018.

[38] Dongxue Zhang, Yang Zheng, Yu Wen, Yujue Xu, Jingchuo Wang, Yang Yu, and Dan Meng. Role-based Log Analysis Applying Deep Learning for Insider Threat Detection. In *Proceedings of the 1st Workshop on Security-Oriented Designs of*

Computer Architectures and Processors - SecArch\textquotesingle18. ACM Press, 2018.

[39] Z. Zheng, Z. Lan, B. H. Park, and A. Geist. System log pre-processing to improve failure prediction. In *2009 IEEE/IFIP International Conference on Dependable Systems Networks*, pages 572–577, June 2009.

[40] Tianqing Zhu, Gang Li, Wanlei Zhou, and Philip S. Yu. Differentially Private Data Publishing and Analysis: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 29(8):1619–1638, August 2017.