An MTech. Dissertation Preliminary Report

titled

POSTING COMMENTS UNDER DIFFERENTIAL PRIVACY

Submitted in partial fulfilment towards the award of the degree of

MASTERS OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

Mr. Nihar Sodhaparmar

P22CS013

Supervisor

Dr. Sankita J. Patel, SVNIT, Surat

2023 – 2024

Department of Computer Science and Engineering

SARDAR VALLABHBHAI NATIONAL INSTITUTE OF TECHNOLOGY, SURAT

SARDAR VALLABHBHAI NATIONAL INSTITUTE OF TECHNOLOGY, SURAT

DECLARATION

I hereby declare that the work being presented in this dissertation preliminaries report entitled "Posting Comments Under Differential Privacy" by me i.e. Mr. Nihar Sodhaparmar, bearing Admn. No: P22CS013 and submitted to the Department of Computer Science and Engineering Sardar Vallabhbhai National Institute of Technology, Surat; is an authentic record of my own work carried out during the period of July 2023 to December 2023 under the supervision of Dr. Sankita J. Patel. The matter presented in this report has not been submitted by me to any other University/Institute for any cause.

Neither the source code there in, nor the content of the project report have been copied or downloaded from any other source. I understand that my result grades would be revoked if later it is found to be so.

_____

(Nihar Sodhaparmar)

C E R T I F I C A T E

This is to certify that the dissertation preliminary report entitled " Posting Comments Under Differential Privacy ", prepared and presented by Mr. Nihar Sodhaparmar, bearing Admn. No: P22CS013 of MTech. - II, Semester - III in Computer Science and Engineering, at Department of Computer Science and Engineering of the Sardar Vallabhbhai National Institute of Technology, Surat is satisfactory.

Certified By

Dr. Sankita J. Patel
Assistant Professor,
Department of Computer Science and Engineering,
Sardar Vallabhbhai National Institute of Technology,
Surat – 395007, Gujarat
India

Jury's Signature

PG Incharge,
M.Tech in Computer Science and Engineering,
SVNIT - Surat

Head,
Department of Computer Science and Engineering,
SVNIT - Surat

Acknowledgments

I am grateful to Dr. Sankita J. Patel, who has been a great advisor from the very beginning. I am thankful to her for valuable discussions and the numerous contributions that she has provided to this work. Without her support and guidance, this work could not have been accomplished. Besides my advisor, I would like to thank my research progress committee members for their encouragement, insightful comments, and suggestions.

I want to thank Dr. Mukesh A. Zaveri, Head of Computer Science and Engineering, SVNIT, Surat, for allowing me to explore research aspects of security and providing infrastructural facilities for my work. I want to thank all the faculties and staff members of the Computer Science and Engineering Department, SVNIT, Surat.

Nihar Sodhaparmar

P22CS013

Abstract

In the realm of social media, the extensive mining of user data has enabled the prediction of sensitive individual attributes, including political and religious beliefs. While such data utilization enhances user experience through personalized services, it also poses potential harm and discrimination, particularly when influencing consequential decisions like activities, behaviors, attributes. The publication of user-generated data poses a risk of compromising individuals' privacy. This work investigates various attribute inference attacks on user data and explores diverse privacy techniques as countermeasures. Furthermore, we propose an approach, "Posting Comments Under Differential Privacy", which generates differentially private comments capable of fooling gender attribute classifiers to protect social media users against gender attribute inference attacks.

Keywords: *Social Networks, Privacy, Machine Learning*

Table of Contents

List of Figures

List of Tables

List of Acronyms

OSN Online Social Network

FOX Fooling with explainations

MALCOM Malicious Comment Generation

LDP Local Differential Privacy

ULDP Utility-optimized Local Differential Privacy

MLDP Metric Local Differential Privacy

LIME Local Interpretable Model-agnostic Explainations

Chapter 1

Introduction

1.1 Motivation

A vast amount of data is produced every day on Online Social Networks (OSN). Millions of peoples across the globe, spanning various age groups uses platforms like Facebook, Instagram, Linked In, Twitter, Reddit and numerous others to facilitate communication, connect with others, share their day-to-day activities, express their views on current political and social events, employment opportunities and many other reasons. Consequently, a tremendous amount of data has been generated and In this process users knowingly or unknowingly disclose the personal information publicly with both friend and strangers. The richness of content within these data, encompasses users' behaviors, relationships, reactions, and other confidential information. This magnifies the risks to individuals' privacy. These data exposes users to tracking, making them extremely vulnerable to potential threats.

User generated data allows researchers and business partners to study and understand individuals on a large scale [4] [5] . Social media data is extensively utilized by a range of data consumers and has become a highly profitable asset for providers of social media services. Companies utilize social media data for the analysis of customer behavior, monitoring public responses to their products, cost-effective delivery of online advertisements, and identification of trends affecting their business. Adversaries can infer the personal information like age, political affiliation and gender of users from the publicly available user's data [6] [7] [8].

In [9] and [10], authors demonstrated the ability to determine sensitive attributes by examining the responses received on the target user's posts (e.g. comments by friends, friends of friends or other users). Data originating from other users is more challenging to control than data created by the users itselfs. For example, a user might avoid using sensitive attributes in their posts. However, it's impractical to expect the same from everyone who responds to their content. This makes the risk of unintentional disclosure of sensitive information through others' reactions more significant.

1.2 Contribution

The goal of this work is to figure out how to use Differential Privacy to generate Differentialy private comment on the OSNs, in order to prevent adversaries from inferring the users' sensitive attributes from the non user generated data(e.g, comments by friends).

To protect OSN users from attribute inference attacks on non user generated data, we propose Posting Comments Under Differential Privacy. We first give dataset (e.g., Facebook pictures comments) to the BERT model and extract the strong features using LIME explainability tool from the reactions which is gender sensitive. By replacing this strong features with the synonym using Differential Privacy method we generate the new Differential Private comment. This newly generated Differential Private comment can hide the gender sensitivity from the classifier models.

1.3 Report outline

The structure of this report is as follow. The second chapter provides the theoretical background of the previous approaches that were trying to solve a similar problem. The third chapter describes the our approach. In the fourth chapter experiments and results is shown. We conclude with future works in fifth chapter.

Chapter 2

Theoretical Background and Literature Survey

Current studies concentrate on three social media applications: establishing personal connections, exchanging contextual information, and receiving personalized services. On online social media, users generate many types of data, such as text, graph, spatiotemporal, and profile attribute data. The authors of [1] categorized the current works into five groups: (1) social graphs and privacy; (2) authors in social media and privacy; (3) profile attributes and privacy; (4) location and privacy; and (5) recommendation systems and privacy. The various combinations of applications and data types that are covered by each category are displayed in Figure 2.1.

		Applications		
		Making Connection with People	Sharing Contextual Information	Receiving Personalized Services
Type of Data	Graph data	• Social graphs & privacy • Profile attributes & privacy	• Profile attributes & privacy	• Recommendation systems & privacy
	Textual Data	• Profile attributes & privacy	• Authors & privacy • Profile attributes & privacy	• Recommendation systems & privacy
	Spatiotemporal Data	• Users location & privacy	• Users location & privacy	• Recommendation systems & privacy
	Profile Attribute Data	• Profile attributes & privacy	• Profile attributes & privacy	• Recommendation systems & privacy

Figure 2.1: Combination of Various Data Types and Applications [1]

The richness of user-generated content, which includes relationships and other personal information, poses a huge risk to people's privacy. In the literature, two categories of information disclosures have been distinguished: identity disclosure and attribute disclosure attacks. Identity disclosure takes place when a person is connected to an individual in a dataset that is made available to the public. Attribute disclosure occurs when disclosed data allows an adversary to derive additional information about a person. Users are still very much at risk from attribute

inference attacks, which include utilizing deep learning or classical learning models to obtain fraudulently private information (such age or gender) from publicly accessible data. Due to user privacy concerns, social media data publishers are required to cleanse user-generated data before making it available to the public in order to safeguard users' privacy. We will begin by describing the Attribute Inference attacks on the user data.

2.1 Attribute Inference Attacks

A user's profile may contain self-identifying attributes, such as age, gender, visited places, and political viewpoits, which could be infer by potential attackers. Social networks typically give users the ability to restrict who can see their attributes, making them only available to friends or friends of friends, in order to protect their privacy. It is also possible that a user can create a profile without expressly sharing any attribute data. As a result, a social network combines user data that is both public and private. Nevertheless, that can be possible privacy attack that is attribute disclosure attack. It uses the publicly available data and infer the missing attributes of the users.

The attacker could be advertisers, data brokers, network service provider itself and cyber criminals. Users' attribute information is used by social network providers and advertisers to deliver more specialized services and advertisements. Data brokers profit by selling personal information to advertising, banks, and insurance providers, among other businesses. Utilizing attribute information, cyber criminals execute spear phishing, backup authentication, and targeted social engineering schemes. This attribute information can be used for gaining various information about users. The majority of studies on attribute inference attacks fall into three categories: friend based, behavior based, and a combination of both.

2.1.1 Friend based Attribute Inference Attacks

Friend based techniques follows the method that two friends are more likely than two strangers to have comparable characteristics. To represent the causal relationships between individuals in the network, In [11] authors first build a Bayesian network using neighbors of a users'. This approach's primary scalability issue stems from the fact that Bayesian inference is insufficient for the millions of members found in social networks.

Zhelva et al. [7] investigate the possibility of users' social connections and group memberships disclosing private attribute information(such as age and location). By using social connections and group data, friend based attribute inference attack infers sensitive characteristics for every user. The authors present a number of algorithms, and among those that just use link information, it was discovered that LINK was the best. This approach represents every

user, u as a binary vector, with each element v having a value of one if u is linked to it, and a length equal to the network's size, or the total number of users. After that, several classifiers are trained on individuals who have public profiles, allowing attributes for users who have private profiles to be deduced. Out of all the techniques that use group information, the GROUP algorithm performed the best. Using either a feature selection methodology (i.e., entropy) or a manual process, this method first chooses the groups that are important to the attribute inference problem. After that, a classifier model is developed using each node's important groups as features.

Mislove et al. present a method that makes advantage of the social connections and community data of users [12]. Their method uses a seed set of individuals with known features as input and then uses link information to identify local communities surrounding this seed set. Next, it makes use of the shared attributes of users within the same community. The remaining users' characteristics are then inferred by this method from the communities in which they participate using fact that community has same attributes. The drawback is that users who have not been assigned to a local community cannot have their attributes inferred using this method.

In another study, Dey et al. [6], concentrate on estimating the ages of Facebook users by taking into account the details of their friends networks. The reverse lookup technique is used in this study to retrieve a partial friends list of each user, even if friends list of users is not entirely accessible to all users. Subsequently, they created an iterative algorithm that determines users' ages by utilizing the ages of friends, friends of friends, and so forth. They also included other publicly available data in their profile, as an example the year they graduated from high school to infer their birth year.

2.1.2 Behavior based Attribute Inference Attacks

Behavior based inference attacks, in contrast to friend based tactics, determines a user's attributes using publicly accessible information about users' activities and publicly available attributes of other users who are similar to user. In [8] present a method that uses user behavior toward movie ratings to infer features (such as gender) about users. Specifically, a vector model with a dimension equal to the number of items is used to represent each user. Each vector element with a non-zero value indicates that the item has been rated by the user.; a zero value indicates no rating at all. Then, they employ various classifiers to determine the ages of users, including logistic regression, SVM, and Naïve Bayes.

In the work [13], authors infers the attributes of the user using the various types of music they like. This method looks for semantic similarities between a user's interests after extracting them. Utilizing an ontologized version of Wikipeda that is relevant to every genre of music, it makes use of topic modeling strategies and builds a semantic interest topic pool for every user.

Next, it is anticipated that a user will share characteristics with those who share their same taste in music.

2.1.3 Friend and Behavior based Attribute Inference Attacks

Another group of methods uses information about user behavior and social links to infer characteristics about the user. In [14], authors first created a social-behavior-attribute network (SBA) using social structure, behaviors and characteristics of users all combined into a single framework. Users, behaviors, or other qualities are represented as nodes in this graph, and the relationships between these attributes are represented as edges. Next, they employ a vote distribution attack (VIAL) model to infer the characteristics of a target user. In the augmented SBA network, VIAL does a designed random walk from the target user to every other user, assigning probabilities to each user based on how structurally similar the user is to the target node in the SBA network. The characteristics of the target user are then inferred using the attribute nodes' stationary probabilities; that is, the target user is assigned the attribute with the highest probability.

Furthermore, there are works that make use of additional information sources including writing style, liked pages, and group subscriptions. But there is a few research done on profile atrribute attacks on data generated by other users. We will see these in the following section.

2.1.4 Non User Generated Data based Attribute Inference Attacks

Non user generated user data refers to information created by individuals other than the primary user. This includes interactions such as reactions made on the user's posts by their friends or friends of friends within the social media network. Essentially, non user generated data captures the responses and engagements originating from the broader social connections of the user, providing a comprehensive view of the user's online interactions and social presence. This type of data plays a crucial role in understanding the user's digital footprint, offering insights into their relationships and interactions within the social media ecosystem.

In [9], author proposes gender inference attacks using meta-data from user-generated pictures on Facebook. The authors evaluate the accuracy of predicting gender using alt-texts, comments, and a combination of both. They also present a selection of sensitive words that can be used to launch successful gender inference attacks.

The authors of [10] presents an online gender inference attack that can detect user gender with high accuracy. This work is extension of work presented in [9]. The attack relies on data not generated by the user, including alt-text generated by Facebook to describe picture content, as well as comments with words and emojis posted by friends, friends of friends, or other users.

The attack utilizes word and emoji vectors along with n-grams to identify features that have the greatest impact on gender inference.

2.2 Attacking Profile Attribute Predictors

Several protective methodologies have been suggested to anonymize social media data generated by users. In this section we go over how we can anonymize user data. So that the profile attribute predictors can not infer the users' attribute.

The authors of [15] discuss the issue of poisoning and evasion attacks on inference models for profile attributes that are dependent on users' liked pages on Facebook. In evasion attacks, the user modifies their profile data by adding or deleting features (Liking/Unliking pages) that influence a certain attribute, that could compromise users' privacy. In Poisoning Attacks, manipulation of the training data is required to reduce the target classifier's overall performance. In Poisoning Attacks, manipulation of the training data is necessary to reduce the overall performance of the target classifier. To reduce the total target classifier performance, 1% of the training data must be manipulated. But generally training data of attribute predictor models is not accessible, particularly for black-box models.

Various papers examine adversarial attacks focused on user-generated data, relatively few address attacks on profile attribute classifiers that uses the reactions generated by other users. In [16], authors suggested a method to generate and appending malicious comments to a news story, which would lead a neural fake news detector to incorrectly identify it as FAKE or REAL news. They propose MALCOM, a Malicious Comment Generation Framework, to mislead fake news detection machine learning models. Malcom uses a conditional text generator to generates malicious comments. Then, they appends this generated comments to the targeted article.

Generating text in an adversarial environment, where the goal is to target machine learning classifiers, is more difficult because text is discrete. A lot of work has gone into generating adversarial samples to target text based machine learning algorithms [17] [18] [19]. The majority of them concentrate on making small adjustments (such additions, deletions, or replacements) to the character [19] [18] or to the word [17]. Even while these techniques work well, they are primarily intended to target static features, such changing a review's content to trick a sentiment analysis classifier models. They are not designed to handle sequential dynamic input, such as comments, where content might be added gradually.

To overcome these autohrs in [2] provide a framework for creating adversarial responses that can successfully target both dynamic sequential inputs (like comments) and static inputs (like alt-text) where the original input shouldn't be changed. With the help of an explainability tool (like LIME) and an initial dataset of Facebook picture reactions, they present a strong framework that can extract powerful adversarial features from reactions. Using these features,

Method	Approach	Limitations
DeepWordBug [19]	Making characters level transformations to fool deep learning classifiers.	Adversial changes can be easily visible because of simple character level transformations
TEXTBUGGER [17]	Making Minimal alterations to the words for attacking deep learning models.	Adversarial changes are easily detectable by individuals due to the minimal alterations to the words.
MALCOM [16]	Generating adversial comments to attack neural fake news detections.	Proposed for only fake news detection, Other users have to add manually adversial comments on fake news article.
FOX [2]	Generating adversial comments to fool gender attribute inference classifiers.	Others users have to add manually comments on the Facebook post.

Table 2.1: Comparison of different approaches to fool classifiers

adversarial responses will be created for fooling black-box classifiers and prevent them from violating the privacy of users of social networks.

Comparison of different approaches to fool classifiers is shown in Table 2.1.

Another work in [3] proposed to hide the sensitive information of text from Machine Learning models using Differential Privacy. They suggest a sanitization-aware pretraining approach. Initially, they employ mechanisms to sanitize public texts, and then they train the Language Model(LM) using the sanitized text. They introduced two token-wise sanitization methods: $SANTEXT$ and $SANTEXT^{+}$.

Drawing inspiration from [2] and [3], we subsequently introduce a novel approach to posting comments under the framework of Differential Privacy. We describe the FOX framework and Sanitization methods using Differential Privacy in the following sections.

2.3 FOX(Fooling With Explations) Framework

In this section, we provide a comprehensive explanation of the FOX framework, describing its components and processes in a step-by-step manner.

Step 1 - Extracting strong adversarial features: We have black box classifier, Which classify the gender of the user. We do not have any idea about the parameters, underlying architecture and training dataset of the black box classifier. Our goal is to determine which features most contributing into each class. Then, apply the LIME explainability tool on the black box classi-

Figure 2.2: User categorized as male before perturbed comment [2]

Figure 2.3: User categorized as female after perturbed comment [2]

fier that extracts the strong adversial features ε^{strong} from the reactions. These features will be utilized to create adversarial reactions, aiming to mislead black-box classifiers and protect the privacy of social media users.

Step 2 - Generating adversarial reactions: Using the strong adversarial features ε^{strong} extracted in above step, FOX model perturb the original comment to fool the black box classifier. The perturbation functions depend on both the feature's reaction type and the original reactions. For example, a comment must be grammatically correct, and a person cannot respond emotionally to the same post in two different ways.

Step 3 - Collaborative privacy protection: If a user is interested in protecting their profile against attribure inference attacks, they can create fake profiles or ask a group of friends to add the FOX-recommended reactions.

2.4 Differential Privacy in Text Sanitization

In this section, we first examine the definition of Differential Privacy. Then, we discuss the two text sanitization techniques proposed in [3].

2.4.1 Defining (Local) Differential Privacy

Assume that all user possesses a document $D = \langle \mathrm{x_i} \rangle_{i=1}^{L}$ of L tokens (each of which could be a character, word, subword or an n-gram), where x_i is drawn from the vocabulary V of size $|V|$. To

protect privacy, user generates a sanitized version $\hat{D}$ by employing a common text sanitization mechanism M on their local devices over D. M replaces replacing each token x_i in D with a substitute token $y_i \in V$.

2.4.1.1 Variants of Local Differential Privacy(LDP)

Consider X and Y represent the input apace and output space, respectively. A randomized mechanism $M : X \rightarrow Y$ is a probabilistic function that produces a random output $y \in Y$ in response to input $x \in X$.

Definition 1 *ϵ − LDP: With a privacy parameter $\epsilon \geq 0$, the mechanism M satisfies ϵ-local differential privacy (ϵ-LDP) under the condition that, for any $x, x', y \in V$,*

$$Pr[M(x) = y] \leq e^{\epsilon} \cdot Pr[M(x') = y] \tag{2.1}$$

From the attacker's perspective, given an observed output y, the likelihoods y are computed from x and x' are same. A smaller ϵ offers better privacy because output distributions are more indistinguishable, but the outputs retain less utility. ϵ-LDP is a very powerful privacy concept because to its uniform protection across all input pairs. However, this is also costly to utility: the output distributions of x and x' must be same, regardless of how unrelated they are. Consequently, a sanitized token y may not capture the semantics of its corresponding input x.

2.4.1.2 LDP in metric spaces

We employ the relaxed notion of Metric-LDP(MLDP) to capture semantics, which was originally proposed for location privacy between two locations (e.g., Manhattan distance).

Definition 2 *MLDP: For any $\epsilon \geq 0$ and a distance metric $d : V \times V \rightarrow R \geq 0$ over V, the mechanism M satisfies MLDP for any $x, x', y \in V$*

$$Pr[M(x) = y] \leq e^{\epsilon \cdot d(x,x')} \cdot Pr[M(x') = y] \tag{2.2}$$

When $d(x, x') = 1$, $\forall x \neq x'$ MLDP and LDP is same. For MLDP, the similarity of output distributions is depends on the corresponding inputs. For MLDP, the metric d must be precisely defined.

2.4.1.3 Incorporating ULDP to increase utility.

Utility-optimized Local Differential Privacy(ULDP) extends the applicability of LDP, initially designed for combining ordinal responses. It makes use of the fact that different inputs have

varying degrees of sensitivity to obtain more benefit. ULDP achieves a privacy guarantee similar to LDP for sensitive inputs by assuming that the input space is partitioned into sensitive and non-sensitive regions.

IN ULDP input space divided into $V_s \subseteq V$ set of sensitive tokens and remaining non sensitive tokens into V_n. The output space divided into protected tokens $V_p \subseteq V$ and remaining into unprotected tokens V_u.

Tokens categorized as sensitive $x \in V_s$ can only be substituted with protected words $y \in V_p$. For non-sensitive words $x \in V_n$, mapping to V_p is allowed, but every substitution $y \in V_u$ must originate from V_n, contributing to enhanced utility.

2.4.1.4 Utility optimized MLDP Notion

Utility optimized MLDP(UMLDP) formulated from the Utility-optimized Local Differential Privacy(ULDP) and Metric Local Differential Privacy(MLDP).

Definition 3 *UMLDP: Given $V_s \cup V_n = V$, along with privacy parameters $\epsilon, \epsilon_0 \geq 0$, and a distance metric $d : V \times V \in \mathbb{R}_0$, the mechanism M satisfies $(V_s, V_p, \epsilon, \epsilon_0)$ - UMLDP, if*

i) for any $x, x' \in V$ and any $y \in V_p$, we have

$$Pr[M(x) = y] \leq e^{\epsilon \cdot d(x,x') + \epsilon_0} \cdot Pr[M(x') = y] \tag{2.3}$$

ii) for any $y \in V_u$, i.e., where $V_u \cap V_p = \phi$, there is an $x \in V_n$ such that

$$\begin{aligned} Pr[M(x) = y] &\geq 0, \\ Pr[M(x') = y] &= 0 \forall x' \in V \setminus \{x\} \end{aligned} \tag{2.4}$$

The UMLDP notion is summarized in Figure 2.4. It shows "invertibility," implying that $y \in V_u$ must be "noise free" and deterministically mapped. Additionally, in generalizing the ULDP definition to $\epsilon \cdot d(x, x')$, we introduce an additive bound $\in_0$ for invertibility, simplifying the derivation of ϵ.

2.4.2 Sanitization Mechanisms

In this section, we go through the two sanitization mechanisms $SANTEXT$ and $SANTEXT^+$.

2.4.2.1 $SANTEXT$ Mechanism

A common step in NLP is to use an embedding model to map semantically comparable tokens to nearby vectors in a Euclidean space. More precisely, an embedding model is defined as an injective mapping $V \rightarrow \mathbb{R}^m$ where m represents the dimensionality. The Euclidean distance

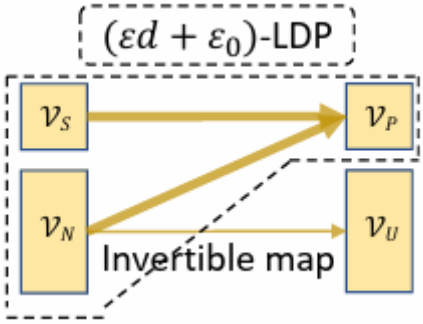

Figure 2.4: Overview of UMLDP notion [3]

between any two tokens, indicated as x and x', is calculated by applying the Euclidean distance metric to their respective embeddings: $d(x, x') = d_{euc}(\phi(x), \phi(x'))$.

Algorithm Base Mechanism SANTEXT

Input: A private document $D = \langle x_i \rangle_{i=1}^{L}$, and a privacy parameter $\epsilon \geq 0$

Output: Sanitized document $\hat{D}$

1 Derive token vectors $\phi(x_i)$ for $i \in [1, L]$;
2 **for** $i = 1, \ldots, L$ **do**
3 Run $\mathcal{M}(x_i)$ to sample a sanitized token y_i with probability
4 **end**
5 Output sanitized $\hat{D}$ as $\langle y_i \rangle_{i=1}^{L}$;

Figure 2.5: Santext algorithm [3]

In figure 2.5, pseudo code of the $SANTEXT$ algorithm is shown. First and foremost, using ϕ convert the each token x of document D into token embeddings. Then, in the second step for each token x run a sanitizing mechanism $M(x)$ with probability

$$Pr[M(x) = y] = C_x \cdot e^{-\frac{1}{2}\epsilon \cdot d_{euc}(\phi(x),\phi(y))} \tag{2.5}$$
$$\text{where } C_x = \left(\sum_{y' \in V} e^{-\frac{1}{2}\epsilon \cdot d_{euc}(\phi(x),\phi(y'))}\right)^{-1}$$

The lower the value of $d_{euc}(\phi(x), \phi(y))$, the more probable y is to replace x. In the last output is sanitized document $\hat{D} = \langle \mathrm{y_i} \rangle_{i=1}^{L}$

2.4.2.2 $SANTEXT^+$ Mechanism

Within the $SANTEXT$ framework, all tokens classified as sensitive, leading to overprotection and subsequent utility loss. In the $SANTEXT^+$ algorithm, vocabulary tokens V divided into V_s

and V_n and focus on the safeguarding V_s.

Algorithm Enhanced SANTEXT$^+$

Input: A private document $D = \langle x_i \rangle_{i=1}^{L}$, a privacy parameter $\epsilon \geq 0$, probability p for a biased coin, and sensitive $\mathcal{V}_S$

Output: Sanitized document $\hat{D}$

Derive token vectors $\phi(x_i)$ for $i \in [1, L]$;
for $i = 1, \ldots, L$ **do**
  if $x_i \in \mathcal{V}_S$ **then**
    Sample a substitution $y_i \in \mathcal{V}_P = \mathcal{V}_S$ with probability
    Run SANTEXT over $\mathcal{V}_S$ and $\mathcal{V}_P$;
  else
    Output $y_i = x_i$ with prob. $(1 - p)$; or $y_i \in \mathcal{V}_P$ with prob.
  end
end
Output sanitized $\hat{D}$ as $\langle y_i \rangle_{i=1}^{L}$;

Figure 2.6: Santext plus algorithm [3]

In figure 2.6, pseudo code of the $SANTEXT^+$ algorithm is shown with $V_s = V_p$ and $V_n = V_u$ shared among all users. In the first step, using ϕ convert the each token x of document D into token embeddings.

Subsequently, if token x belongs to V_s, sample the substitution of y from V_p according to the probability specified in the $SANTEXT$ algorithm. This is the same as employing SANTEXT across V_s and V_p.

For $x \in V_n$, flip a bias coin. If the result is $(1 - p)$ outputs y as x. Otherwise, samples $y \in V_p$ with probability

$$Pr[M(x) = y] = p \cdot C_x \cdot e^{-\frac{1}{2}\epsilon \cdot d_{euc}(\phi(x),\phi(y))} \tag{2.6}$$

$$\text{where } C_x = \left(\sum_{y' \in V_p} e^{-\frac{1}{2}\epsilon \cdot d_{euc}(\phi(x),\phi(y'))} \right)^{-1}$$

In the last output is sanitized document $\hat{D} = \langle \mathrm{y_i} \rangle_{i=1}^{L}$

Chapter 3

Implementation Methodology

3.1 Proposed Approach

We propose approach Posting Comments Under Differential Privacy, which given first dataset X (e.g., comments of Facebook users) and an explainability tool such as LIME. Then, extracts strong adversarial features ε^{strong} from comments. These strong features is used to convert comments into Differential Private comments to mislead classifiers to avoid them from compromising the privacy of users. In the below we describe the steps, and notations listed in Table 3.1.

Step 1 - Extracting strong features: We have dataset X (e.g., comments of Facebook users). For each instance $x \in X$, we have set of features F (e.g., words of comments can be set of features).

For a given set of classes $Y = \left\{y_1, y_2, ..., y_{|y|}\right\}$ a classifier $h : X \rightarrow [0, 1]^{|Y|}$ maps x to a vector $h(x) = \left[p_1, p_2, ..., p_{|y|}\right]$, where $h(x)[i] = p_i$ represents the probability that x belongs to class y_i. The class label of x, that is the class y_i with the highest probability p_i, is denoted by $h_l(x)$. We have to determine the most contributing features.

Let us define the $\mathcal{E}(E, x, h)$ function, which accepts Explainaibility tool E, an instance of dataset x and classifier h as input and returns the strong features for the instance x as output.

Symbol	Description
X	Dataset of Facebook picture comments
x	Instance of a dataset (e.g., comment)
F	Features set of x
Y	Set of classes
$h(.)$	Classifier
$h_l(.)$	Class label returned by classifier
$E(.)$	Explainability tool
ε^{strong}	Strong features

Table 3.1: Notations

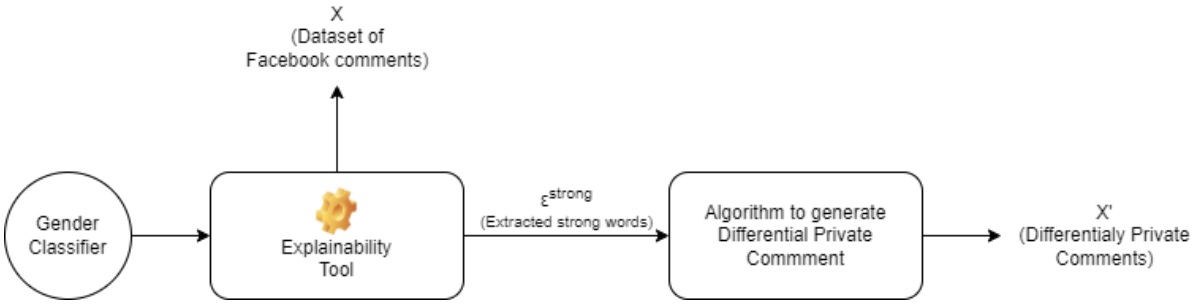

Figure 3.1: Generating Differentialy Private Comments

Now, extracting strong words from all dataset given as below,

$$\varepsilon^{strong} = \bigcup_{x \in X} \mathcal{E}(E, x, h)$$

Step 2 - Generating Differentialy Private Comments: After constructing ε^{strong}, the next step involves generating Differentialy Private comments to mislead classifier h. For generating Differentialy Private comments we use the $SANTEXT^{+}$ algorithm proposed in [3]. Extracted strong words are considered as Sensitive tokens V_s, while other tokens considered as non sensitive tokens V_n for the $SANTEXT^{+}$ algorithm. For the substitution of words we use the standard database as vocabulary. Suppose, standard database tokens are represented as V_g. Then protected tokens are $V_p = V_g - V_s$, while remaining tokens are considered as unprotected tokens V_u. Algorithm to generate Differentialy comment shown in 1.

Algorithm 1 Differentialy Private Comment Generator Algorithm

Input: x instance of database X, where $x = \langle \mathrm{F_i} \rangle_{i=1}^{L}$, ε^{strong} features as V_s, probability p for biased coin, privacy parameter $\epsilon \geq 0$

Output: Differentialy Private Comment $\hat{x}$

Derive token vectors $\phi(F_i)$ for $i \in [1, L]$;
for $i = 1, ..., L$ **do**
  if $F_i \in V_s$ **then**
    Sample a substitution $(F')_i \in V_p$ with probability given in Eq. (2.5);
  else
    Output $(F')_i = F_i$ with probability $(1 - p)$;
    or $(F')_i \in V_p$ with probability in Eq. (2.6);
  end if
end for
Output Differential Private Comment $\hat{x} = \langle (\mathrm{F'})_\mathrm{i} \rangle_{i=1}^{L}$

Step 3 - Checking similarity of Original Comment and Differentialy Private Comment: In this step compare the similarity between Original Comments and Differentialy Comment to check how much meaning preserved in the Differentialy Private comment.

Chapter 4

Experiments

4.1 Dataset

Our experiments were conducted on a dataset obtained from Facebook by the authors referenced in [2]. Dataset contains only English language comments. All the comments are generated by other users only. It comprises 4,509 user profiles along with their associated pictures. There is total 190,104 number of comments in dataset. From these 102,884 are comments for male and 87,220 are comments for female.

4.2 Experimental Setup

Step 1 - Extracting strong features: We fine tuned the BERT to infer gender from the comments. Specifically, the utilized BERT model is the 'bert-base-cased' variant. Fine tuned model architecture consists of a BERT base model followed by several fully connected layers with dropout for gender classification. Before fine tune the BERT model we pre-processed the dataset. We removed stopwords from the dataset and convert repeated symbols into one symbol. We split this cleaned dataset into 80:20 for the training dataset and testing dataset. Using training dataset we fine-tuned the BERT classifier model. Now, we have the trained classifer h. Next, using BERT classifier h and LIME explainaibility tool we extract the strong words ε^{strong}.

Step 2 - Generating Differentialy Private Comments: Extracted strong words ε^{strong} generated in above step is considered as sensitive tokens V_s and other tokens as non sensitive tokens V_n. For the vocabulary we use the Glove Wiki Giga Word as standard database. Now, protected tokens are $V_g - V_s$ and remaining tokens from Glove Wiki Giga Word as standard database is considered as unprotecccted tokens. Then, we generate the Differentialy Private comments for all comments in dataset using algorithm 1.

Step 3 - Checking similarity of Original Comment and Differentialy Private Comment: For comparing similarity between original comments and Differential comment, we used bert base nli mean tokens model. Bert base nli mean tokens model convert the senteces into 768

	Accuracy of BERT Model
On Original Comments	80.86%
On Differentialy Private Comments	41.36%

Table 4.1: Accuracy of BERT Model

dimensional vector. We give the Original Comment(x) and Differentialy Comment($\hat{x}$) inputs to the bert base nli mean token model, then get the sentence vectors S_x for Original Comment and $S_{\hat{x}}$ for Differential Private comment. Now, we check the similarity between S_x and $S_{\hat{x}}$ using cosine similarity. If similarity score is close 1 then two comments have similar meanings.

4.3 Results

For the BERT classifier h on the original training dataset and testing dataset we got the 80.86% and 75% accuracy respectively. After generating Differentialy Private we provide these comments to BERT classifier and it's accuracy decreases. It gives the accuracy 41.36% on the Differentialy Private Comments. While checking similarity between Original Comments and Differentialy Comments we get the average similarity score 0.9059. We got the 85.26% times similarity score greater than 0.8 and 66.55% times similarity score greater than 0.9. In similarity checking we got 0.0941 Mean Absolute Error and 0.0216 Mean Squared Error.

From the above results we can say that we can successfully able to fool classifier using our Posting Comments Under Differentialy Private approach. Also we can say that most of the time it preserving the meaning of generated Differentialy Private comment.

Chapter 5

Conclusion and Future Work

In this research, we proposed the approach of Posting Comments Under Differential Privacy to protect social media users against attribute inference attacks, specifically focusing on gender classification in comments. We utilized a BERT classifier fine-tuned for gender inference and extracted strong features using LIME as an explainability tool. These strong features were then employed to generate differentially private comments, intending to mislead the classifier while preserving the meaning of the comments.

Our experiments, conducted on a dataset obtained from Facebook, demonstrated promising results. The BERT classifier achieved 80% accuracy on the original training dataset and 75% on the testing dataset. However, when applied to differentially private comments, the accuracy decreased significantly to 41.36%, indicating the effectiveness of our approach in confusing the classifier.

Moreover, the similarity analysis between original and differentially private comments revealed an average similarity score of 0.9059. This suggests that, despite the intentional perturbations introduced to mislead the classifier, the differentially private comments retained a substantial portion of their original meaning. The majority of the similarity scores exceeded 0.8(85.26%) and 0.9(66.55%), further supporting the claim that our approach successfully preserves the semantic content of the comments.

In conclusion, the Posting Comments Under Differential Privacy approach demonstrates promise in protecting user privacy on social media by misleading attribute classifiers. While achieving a trade-off between privacy and utility, our method successfully creates differentially private comments that, most of the time, retain the meaning of the original comments. These findings underscore the potential of leveraging privacy-preserving techniques to safeguard users in the face of attribute inference attacks on social media platforms.

For future work, our intention is to extend the applicability of our Posting Comments Under Differential Privacy approach by testing it on various machine learning models apart from BERT. Exploring its effectiveness and adaptability across different classifiers will provide insights into the generalizability of our method. Additionally, we aim to enhance the sentence similarity analysis by incorporating context words during the modification of strong features.

Bibliography

[1] G. Beigi and H. Liu, "A survey on privacy in social media: Identification, mitigation, and applications," *ACM/IMS Trans. Data Sci.*, vol. 1, no. 1, mar 2020. [Online]. Available: https://doi.org/10.1145/3343038

[2] N. Belhadj-Cheikh, A. Imine, and M. Rusinowitch, "Fox: Fooling with explanations: Privacy protection with adversarial reactions in social media," in *2021 18th International Conference on Privacy, Security and Trust (PST)*. IEEE, 2021, pp. 1–10.

[3] X. Yue, M. Du, T. Wang, Y. Li, H. Sun, and S. S. Chow, "Differential privacy for text analytics via natural text sanitization," *arXiv preprint arXiv:2106.01221*, 2021.

[4] G. Beigi, "Social media and user privacy," *arXiv preprint arXiv:1806.09786*, 2018.

[5] H. Gao, J. Hu, T. Huang, J. Wang, and Y. Chen, "Security issues in online social networks," *IEEE Internet Computing*, vol. 15, no. 4, pp. 56–63, 2011.

[6] R. Dey, C. Tang, K. Ross, and N. Saxena, "Estimating age privacy leakage in online social networks," in *2012 proceedings ieee infocom*. IEEE, 2012, pp. 2836–2840.

[7] E. Zheleva and L. Getoor, "To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles," in *Proceedings of the 18th international conference on World wide web*, 2009, pp. 531–540.

[8] U. Weinsberg, S. Bhagat, S. Ioannidis, and N. Taft, "Blurme: Inferring and obfuscating user gender based on ratings," in *Proceedings of the sixth ACM conference on Recommender systems*, 2012, pp. 195–202.

[9] B. Alipour, A. Imine, and M. Rusinowitch, "Gender inference for facebook picture owners," in *Trust, Privacy and Security in Digital Business: 16th International Conference, TrustBus 2019, Linz, Austria, August 26–29, 2019, Proceedings 16*. Springer, 2019, pp. 145–160.

[10] B. A. Pijani, A. Imine, and M. Rusinowitch, "Online attacks on picture owner privacy," in *Database and Expert Systems Applications: 31st International Conference, DEXA 2020, Bratislava, Slovakia, September 14–17, 2020, Proceedings, Part II 31*. Springer, 2020, pp. 33–47.

[11] J. He, W. W. Chu, and Z. Liu, "Inferring privacy information from social networks," in *International Conference on Intelligence and Security Informatics*. Springer, 2006, pp. 154–165.

[12] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel, "You are who you know: inferring user profiles in online social networks," in *Proceedings of the third ACM international conference on Web search and data mining*, 2010, pp. 251–260.

[13] A. Chaabane, G. Acs, M. A. Kaafar *et al.*, "You are what you like! information leakage through users' interests," in *Proceedings of the 19th annual network & distributed system security symposium (NDSS)*. Citeseer, 2012.

[14] N. Z. Gong and B. Liu, "Attribute inference attacks in online social networks," *ACM Transactions on Privacy and Security (TOPS)*, vol. 21, no. 1, pp. 1–30, 2018.

[15] Y. Alufaisan, Y. Zhou, M. Kantarcioglu, and B. Thuraisingham, "Hacking social network data mining," in *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*. IEEE, 2017, pp. 54–59.

[16] T. Le, S. Wang, and D. Lee, "Malcom: Generating malicious comments to attack neural fake news detection models," in *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2020, pp. 282–291.

[17] J. Li, S. Ji, T. Du, B. Li, and T. Wang, "Textbugger: Generating adversarial text against real-world applications," *arXiv preprint arXiv:1812.05271*, 2018.

[18] A. Mathai, S. Khare, S. Tamilselvam, and S. Mani, "Adversarial black-box attacks on text classifiers using multi-objective genetic optimization guided by deep networks," *arXiv preprint arXiv:2011.03901*, 2020.

[19] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi, "Black-box generation of adversarial text sequences to evade deep learning classifiers," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 50–56.