User Identity Linkage: Data Collection, DataSet Biases, Method, Control and Application

by
Rishabh Kaushal

Under the supervision of Prof. Ponnurangam Kumaraguru

Indraprastha Institute of Information Technology - Delhi
October, 2020

User Identity Linkage: Data Collection, DataSet Biases, Method, Control and Application

by

Rishabh Kaushal

Submitted
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

to the

Indraprastha Institute of Information Technology - Delhi
October, 2020

Certificate

This is to certify that the thesis titled "**User Identity Linkage: Data Collection, DataSet Biases, Method, Control and Application**" being submitted by Rishabh Kaushal to the Indraprastha Institute of Information Technology Delhi, for the award of the degree of Doctor of Philosophy, is an original research work carried out by him under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree/diploma.

October, 2020

Prof. Ponnurangam Kumaraguru, 'PK'

Indraprastha Institute of Information Technology Delhi

New Delhi 110 020

Acknowledgments

I write with the deepest sense of gratitude to my advisor, Prof. Ponnurangam Kumaraguru (PK), who has taught me, not just the research approach and the technical skills, but also important life lessons through his actions. He has always been extremely supportive and considerate of my contributions during my PhD life. To have PhD advisor like him has been my privilege and good fortune. His valuable inputs and guidance throughout my PhD life have been very helpful. The work in this thesis wouldn't have been possible without his continuous support.

In addition, I obtained a lot of support from my monitoring committee members: Dr. Arun Balaji Buduru, and Dr. Rajiv Ratn Shah. I thank them for providing me valuable suggestions and comments, which helped immensely in the presentation of my work. I am extremely thankful to Prof. Anupam Joshi for the insightful suggestions and research directions in the early stages of my PhD life when I was exploring research problems to work upon.

I would like to thank my co-authors Paridhi Jain, Prateek Dewan, Srishti Chandok, Shubham Singh, Shubham Gupta, Nalin Gupta, Vasundhara Ghosh and Chetna Sharma who played their roles in contributing to the work and providing important technical inputs. I thank Ananya Aggarwal for developing a web service of clonaware. Besides the above, I am extremely thankful to Dr. Paridhi Jain for helping me to understand the problem of identity resolution, explaining the nuances in it, and future directions. I am grateful for the support extended to me by my extremely talented and supportive lab mates Anupama Aggarwal, Dr. Prateek Dewan, Dr. Niharika Sachdeva, and Dr. Srishti Gupta at Precog Research Group.

I am also thankful to faculty, students, and staff at Indira Gandhi Delhi Technical University for Women (IGDTUW) for providing support.

This work would not have been possible without the constant support of my family. My wife Pratibha has provided insightful suggestions, she has been extremely caring, supportive, and provided me with the much needed environment for research at home. My son Ridharv with his playful activities has been a source of energy. I am grateful to my father Sh. Ashwani Kumar Kaushal and my mother Smt. Praveen Kaushal for my upbringing, encouraging me to follow the righteous path even if it has hardships ever since the formative years of my life. I am extremely thankful to my younger sister Shikha for the thoughtful discussions and the new perspectives that she brings with her.

Abstract

Online Social Networks (OSNs) have become popular platforms for online users. Users typically register and maintain their accounts (user identities) across different OSNs to share a variety of content and remain connected with their friends. Consequently, linking user identities across OSN platforms, referred to as *user identity linkage* becomes a critical problem. Solving this problem enables us to build a more comprehensive view of a user's activities across OSNs, which is highly beneficial for targeted advertisements, recommendations, and many more applications. In this thesis, we define the core research statement as follows. *Computational approaches can be proposed for the analysis of data collection methods, investigation of biases in identity linkage datasets, linkage of user identities across social networks, control-ability of user identity linkage, and application of user identity linkage solution to solve extraneous problems.* To that end, we make contributions starting from the computational interventions at the data collection stage, methodology stage, and finally at the implication (privacy and security) stage, for the problem of user identity linkage, as outlined below.

The collection of ground truth data comprising user identity pairs belonging to the same individual is a very important first step. Specifically, we provide a detailed methodology of five methods, namely Advanced Search Operator (ASO), Social Aggregator (SA), Cross-Platform Sharing (CPS), Self-Disclosure (SD), and Friend Finding Feature (FFF) for data collection. Taken together, we collect linked identities of 208,120 individuals distributed across 43 different OSNs. Subsequently, we compare these methods, both qualitatively and quantitatively. Furthermore, we find that user identity datasets obtained from different data collection methods have inherent biases driven by user behaviors. For instance, we find that user identities collected through SD method have more similar usernames and display names than those user identities collected through CPS method. We detect, quantify, and mitigate these dataset biases. We study these biases on more than 1 million user identity pairs obtained by leveraging two user behaviors, namely cross-posting and self-disclosure. We find that biases manifest in the form of lexical differences in user-generated content, particularly in usernames and display names configured by users. These behavioral biases lower down the performance (precision and recall) of learning models by 5-20%. Inspired by discrimination measurement metrics, we propose and implement a framework to quantify the extent of these biases and find that 15-20% of test data get affected. Lastly, we propose an approach to mitigate these biases in the dataset.

At the level of methodology, we propose a node embedding based framework, referred to as *NeXLink* that leverages state-of-the-art node embedding algorithms to learn projections of cross-network linkages (CNLs). A CNL is a pair of user identities across two different social networks belong to the same individual. The NeXLink framework's goal is to project CNLs into an embedding space such that user pairs across OSNs that belong to the same individual are closer than other pairs. Our modular and flexible node embedding framework referred to as *NeXLink*, which comprises three steps. First, we obtain local node embeddings by preserving the local structure of nodes within the same social network. Second, we learn the global node embeddings by preserving the global structure, which is present in the form of common friendship exhibited by nodes involved in CNLs across social networks. Third, we combine the local and global node embeddings, which preserve

local and global structures to facilitate the detection of CNLs across social networks. We evaluate our proposed framework on an augmented (synthetically generated) dataset of 63,713 nodes & 817,090 edges and a real-world dataset of 3,338 Twitter-Foursquare node pairs. Our approach achieves an average hit rate of 98% and 88% in augmented and real-word dataset, respectively, for detecting CNLs across social networks and significantly outperforms previous state-of-the-art methods.

From a privacy perspective, linking user identities across OSNs could potentially result in information leak, particularly for privacy-conscious users. Therefore, we develop a system, which we refer to as *Nudging Nemo*, to help users understand the factors leading to the linkage of their identities across OSNs. Besides, our system helps users control the linkability of their identities across OSN platforms. We evaluate the nudge's effectiveness by conducting a controlled user study on privacy-conscious users who maintain their accounts on Facebook, Twitter, and Instagram. Outcomes of user study confirmed that the proposed framework helped most of the participants to make informed decisions, thereby preventing inadvertent exposure of their personal information across social network services.

Lastly, we apply the methods to detect identities belonging to the same person across social networks onto the single social network scenario to find *identity clones*, who are those users who create their online identities impersonating a real user (victim). We investigate behaviors of clones of celebrities and find them indulging in misbehaviors like spreading indecency, misinformation, and many others.

Keywords: Identity Linkage, Online Social Networks, Data Collection, Dataset Biases, Graph Embedding, Nudge Design, Applications.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Since the time immemorial, human beings have always looked for different ways to socialize [60] with each other. With the scientific discoveries, inventions, and technological advancements, human civilizations [152] have evolved. In the last few decades, the information technology revolution [38, 46] in general, and Web 2.0 [113], in particular, has given way to the emergence of several online platforms, referred to as Online Social Networks (OSNs). Given the universal desire of humans to connect, these new-age platforms have become a popular medium for their users to socialize in the online world. Users post, share and view content on these OSN platforms. Novel ways are being devised by OSNs to attract users to use their platforms, we mention a few popular ones here.

Facebook [127], with over 2.2 billion monthly active users, is the most popular platform. With 330 million registered users, Twitter [69] is a fast-paced, concise, and easy way to connect with one's audience. LinkedIn focuses on business and professional communities [120], with 660 million user base. YouTube is the leading video-sharing platform with 2 billion monthly active users [64] viewing or sharing video content. In terms of content being offered, some OSN platforms offer video (like YouTube and Vimeo), some offer image (like Instagram and Flickr) and others offer a combination of text with image & video (Facebook and Twitter). Users view and engage with the content of their friends. In terms of friend connections, some OSN platforms are used for the professional network (like LinkedIn) while others for close and family friends in general (like Facebook) [153]. Owing to privacy concerns, some social networks [27] (like Whisper and Reddit) allow users, by design, to post messages anonymously. Some ephemeral social networks [9] (Snapchat) keep user content temporarily for some time and then remove it. In short, there are many different kinds of social networks offering different services to users.

Given that many OSNs are offering different services, it is natural for users to create accounts (referred to as *user identities*) on more than one OSN platform. As per Pew Research Center [139],

more than half of online users (56%) use more than one Online Social Media[1] (OSM) platform, a trend that has been consistent in the past few years. Furthermore, among these users who use more than one OSM platform [26], the average number of social media accounts that each such user maintains has increased from 4.3 to 7.6 from the year 2013 to 2017. In the research community as well, Liu et al. [88] find that an individual joins 3.99 social networks on average. Fig 1.1 depicts users joining multiple OSN platforms leading to the problem of *User Identity Linkage* (UIL).

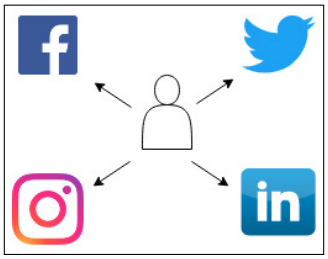


Figure 1.1: A typical scenario where the same user has accounts (referred to as user identities) across many social networks.

We define UIL as a problem of finding user identity on target OSN when that user's identity is known on source OSN, as depicted in Fig 1.2. We refer to the user identities on different OSN platforms belonging to the same person as *linked user identities*.

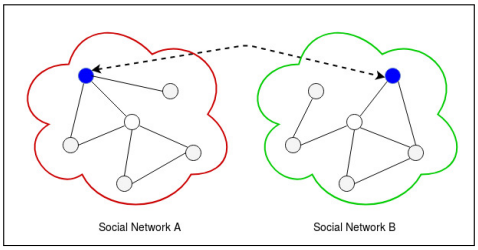


Figure 1.2: Two social networks A and B are given along with users (represented by circles) in each of them. The goal is to link (represented by dotted lines) user identities belonging to the same person across the two social networks.

More formally, we define the UIL problem as follows.

Definition 1.0.1 *Given two user identities I_a and I_b on OSNs a and b, respectively, the goal is to learn a function F, which tells whether I_a and I_b belong to the same person or not.*

$$F(I_a, I_b) = \begin{cases} 1, & \text{if } I_a \text{ and } I_b \text{ belong to the same person.} \\ 0, & \text{otherwise} \end{cases} \tag{1.1}$$

[1]More specifically, OSN is referred to platforms which emphasize on networking among users, and OSM platforms focus on the content. However, we use the term OSN and OSM, interchangeably in this thesis.

Prior works [23, 42, 93, 122, 140, 175] learn the function in two broad ways. The *first* approach is to create handcrafted features derived from the user's profile, content, and network. These features are then fed as input to the machine learning algorithms, as shall be explained in Section 2.3. The *second* approach is to learn user identity representation, as discussed in Section 2.4, in the form of an embedding vector and then apply machine learning algorithms on the learned embeddings.

We make the following contributions to the different paradigms of computer science in this thesis.

- Analysis: Selecting the most suitable approach to collect user identities belonging to the same person is the most important first step. Therefore, we present a detailed quantitative and qualitative comparative analysis of different data collection methods to collect linked user identities.

- Design: We make three design contributions.
  - Given that data collection methods rely on user behaviors, therefore, behavioral biases get manifested in the user identity datasets. We design an approach inspired by the situational testing method adopted by discrimination studies to detect and quantify biases in identity linkage datasets.
  - We solve the UIL problem by designing a modular and flexible *NeXLink* framework, which is based on the concept of node embedding. Nodes that represent user identities belonging to same person are closer in the embedding space than other nodes.
  - Owing to the privacy implications of linking user identities, we design a soft paternalistic nudge that helps users control the linkability of their identities across OSN platforms.

- Applied: Given the ease of account creation and the availability of personally identifiable information (PII), malicious users impersonate real users and create clone identities. We apply the approach used to solve the UIL solution to detect clone identities that belong to the same person within the same OSN platform.

1.1 Motivation & Use Cases

Linking user identities on many OSNs is significant for many reasons. *Firstly*, it provides a more comprehensive description of the user by aggregating [41] user information on different OSNs, in terms of profile attributes, content posted or engaged, and network (friends) maintained. This comprehensive view of users facilitates a better understanding of users' interests, thereby enabling better personalization and recommendations [8, 63, 78, 116, 185] and better social media profiling of an individual for surveillance purposes. *Secondly*, it helps in predicting user behaviors, network dynamics, and information diffusion on a relatively newer OSN platform based on user behaviors in well established existing OSNs [63], an issue commonly referred to as the cold-start problem.

Thirdly, user migration from one OSN platform to another OSN can be investigated [93, 116]. Besides the motivations outlined above, we also present a few indicative cases where solving the UIL problem would be beneficial.

- **Background Checks**: Consider an HR manager who needs to perform background checks [42] for the prospective employees in her organization to understand their capabilities better [189]. To that end, she needs to gather users' behavior across different OSN platforms, which require a linkage of user identities across these platforms.
- **Law Enforcement**: Quite often, law enforcement officials investigate an online crime committed by an accused on a social network. The accused may not have left any evidence in the form of digital footprints on the specific social network under investigation. However, the accused may have left behind questionable behaviors on other social networks presuming that identities could not be linked. So, linking the accused's identity across social networks would help investigate the crime incident by collecting evidence against him. [189].
- **Age Verification**: One of the challenging problems in OSN platforms is that of age verification [175]. Users may misrepresent their age information. So, if inconsistencies and conflicting information is retrieved from the linked identities belonging to the same user, then these users can be further investigated. On the other hand, the user would be more trustworthy if the information is consistent across OSNs [116].
- **Solve Traditional Problems**: Consider social network researcher who solves traditional problems like influence estimation and user expertise estimation on a single OSN platform [93]. However, with aggregated user information obtained after linking user identities across OSNs, he can solve the same problems more effectively.
- **Digital Marketing**: Consider a digital marketing firm who can save marketing revenues for their clients by ensuring that if a user has been targeted in one social network, then the same user need not be targeted in another social network [8].

There are many more such use cases where the solution to the UIL problem is applicable, however, we move on to outline the challenges.

1.2 Key Challenges

There are several challenges in solving the user identity linkage problem, which we enlist in this section. By design, each website has its own sign-up. There is no single sign-up option for multiple OSN platforms [175], which could have made solving the UIL problem trivial. At the very outset,

making a good choice of data collection method to collect ground truth positive identity pairs belonging to the same person is critical [73]. It involves leveraging specific user behaviors which facilitate the collection of user identities across social networks. The user identity datasets thus obtained from a particular data collection method often suffers from biases [74] because the data collection method relies on user behaviors. The methods employed to solve the UIL problem rely upon user activities on the OSN platform in terms of their profile settings, content posted, engagements, and network of friends maintained. User activities are quite unpredictable and unreliable. The amount of profile information made available by user varies from person to person and from platform to platform. Some users are open to disclosing most of their profile attributes, others would be skeptical, and some would deliberately enter misleading information [116]. Not just users, some OSN platforms ask for more profile attributes than others. Given the diversity of content offered by different OSNs, the user-generated content (*what*) and the content generation patterns in terms of time (*when*) and location (*where*) varies substantially from one user (*who*) to another across OSN platforms [133]. The decisions to accept friend request varies across different OSN platform. Some of these platforms help build professional networks (like LinkedIn) where users connect to like-minded professionals, not necessarily their acquaintances. Given these diversities, the network (friend circle) maintained by users often also varies from one OSN to another [187]. All of this means that user identities manifested through their profile, content and network information are quite diverse across different OSN platforms. Therefore, the methods to link user identities need to be robust, modular, flexible, and adaptable to these diversities [75]. From the user's privacy perspective, it is challenging to strike a balance between two extreme ends of the spectrum of users, those who are privacy-fundamentalists versus those who are privacy-unconcerned [81]. There are privacy implications for users who want to keep their identities separate in different OSNs [141], while on the other hand, there are users who want to keep similar identities across OSNs to project same user profile. Keeping these challenges at the back of our mind, we present our thesis statement ahead.

1.3 Thesis Statement

In this thesis, we focus on the following research statement.

Computational approaches can be proposed for the analysis of data collection methods, investigation of biases in identity linkage datasets, linkage of user identities across social networks, control-ability of user identity linkage, and application of user identity linkage solution to solve extraneous problems.

To address this statement, we focus on the problem of user identity linkage from multiple perspectives, our contributions are enlisted in the next section.

1.4 Thesis Contribution

The main contributions of the thesis are:

Methods for User Profiling Across Social Networks: Users have their accounts across multiple Online Social Networks (OSNs). To obtain a comprehensive view of user activities, an essential first step is to link user accounts (identities) belonging to the same individual across OSNs. To this end, we provide a detailed methodology of five methods useful for user profiling, which we refer to as Advanced Search Operator (ASO), Social Aggregator (SA), Cross-Platform Sharing (CPS), Self-Disclosure (SD) and Friend Finding Feature (FFF). Taken all these methods together, we collect linked identities of 208,120 individuals distributed across 43 different OSNs. We compare these methods quantitatively based on social network coverage and the number of linked identities obtained per-individual. We also perform a qualitative assessment of linked user data, thus obtained by these methods, on the criteria of completeness, validity, consistency, accuracy, and timeliness.

Investigation of Biases in Identity Linkage DataSets: Prior works link user identities across OSNs using two steps. First, they collect ground truth datasets of user identities across social networks belonging to the same individuals and then in the second step, they build a machine learning model whose features are derived from user identities. Data collection methods rely on user behaviors on different social networks, and as a consequence, behavioral biases get manifested in the user identity datasets. We perform a detailed investigation into these dataset biases, a work which has mostly remained under-explored in the identity linkage research. More specifically, we characterize, detect, and quantify behavioral biases in these datasets. We find that biases manifest in the form of lexical differences in user-generated content, particularly in usernames and display names configured by users. For quantification, we design an approach inspired by the situation testing framework [99] adopted by discrimination studies to quantify biases in identity linkage datasets.

NeXLink: Node Embedding Framework to solve UIL problem: Users create accounts on multiple social networks. A pair of user identities across two different social networks belonging to the same individual is referred to as Cross-Network Linkages (CNLs). We model the social network as a graph to explore the question, whether we can obtain effective social network graph representation such that node embeddings of users belonging to CNLs are closer in embedding space than other nodes, using only the network information. We propose a modular and flexible node embedding framework referred to as NeXLink, which comprises of three steps. First, it obtains local node embeddings by preserving the local structure of nodes within the same social network. Second, it learns the global node embeddings by preserving the global structure, which is present in the form of common friendship exhibited by nodes involved in CNLs across social networks. Third,

it combines the local and global node embeddings, which preserve local and global structures to facilitate the detection of CNLs across social networks. We evaluate our proposed framework on an augmented (synthetically generated) dataset of 63,713 nodes & 817,090 edges and real-world dataset of 3,338 Twitter-Foursquare node pairs. Our approach achieves an average hit rate of 98% and 88% in augmented and real-word dataset, respectively, for detecting CNLs across social networks and significantly outperforms previous state-of-the-art methods.

Nudging Nemo: Helping Users Control Linkability across Social Networks: Numerous techniques to link user identities across different OSNs have been proposed. However, this linking poses a threat to the users' privacy; users may or may not want their identities to be linkable across networks. We propose *Nudging Nemo*, a framework that assists users in controlling the linkability of their identities across multiple platforms. Nudging Nemo has two components, (1) a linkability calculator, which uses state-of-the-art user identity linkage techniques to compute a normalized linkability measure for each pair of social network platforms used by a user, and (2) a soft paternalistic nudge, which alerts the user if any of their activity violates their preferred linkability. We evaluate the effectiveness of the nudge by conducting a controlled user study on privacy-conscious users who maintain their accounts on Facebook, Twitter, and Instagram. Outcomes of user study confirm that the proposed framework helped most of the participants to make informed decisions, thereby preventing inadvertent exposure of their personal information across social network services.

Detecting of Clone Identities in Online Social Networks: (OSNs) are simple to facilitate users to join the OSN sites. Alongside, Personally Identifiable Information (PII) of users is readily available on- line. Therefore, it becomes trivial for a malicious user (attacker) to create a spoofed identity of a real user (victim), which we refer to as clone identity. We leverage the identity linkage approaches to detect clone identities and then analyze clone identities to extract an exhaustive set of 40 features based on posting behavior, friend network, and profile attributes. These clone identities ride on the credibility and popularity of celebrities to gain engagement and impact. We characterize their behavior as benign and malicious. On detailed inspection, we find benign behaviors are either to promote the celebrity which they have cloned or seek attention, thereby helping in the popularity of celebrity. However, on the contrary, we also find malicious behaviors (misbehaviors) wherein clone celebrities indulge in spreading indecent content, issuing advisories and opinions on contentious topics. We evaluate our approach on a real social network (Twitter) by constructing a machine learning based model to automatically classify behaviors of clone identities, and achieve accuracies of 86%, 95%, 74%, 92% & 63% for five clone behaviors corresponding to promotion, indecency, attention-seeking, advisory, and opinionated.

1.5 Organization of Thesis

We organize this document as follows. In Chapter 2, we present the prior work done in solving the problem of user identity linkage from different perspectives. We explain prominent data collection approaches for collecting linked user identities across social networks in Chapter 3, followed up a comparative study. Subsequently in Chapter 4, we investigate dataset biases in the user identity datasets. In particular, we leverage the approaches from the literature of discrimination studies to detect, quantify, and mitigate these biases. In Chapter 5, we provide details of the node embedding based framework, referred to as *NeXLink* that leverages state-of-the-art node embedding algorithms to project cross-network linkages into an embedding space such that user pairs across OSNs that belong to the same individual are closer than other pairs. Given that linking user identities across OSNs have privacy implications, therefore in Chapter 6, we develop a system, which we refer to as *Nudging Nemo*, to help users understand the factors leading to the linkage of their identities across OSNs. We also discuss how our system helps users control the linkability of their identities across OSN platforms. In Chapter 7, we apply methods to link user identities across social networks in the context of a single social network scenario to detect *identity clones*, the users who create their online identities impersonating a real user (victim). Lastly in Chapter 8, we conclude the thesis by providing details of the implications, limitations and future work.

1.6 Thesis Publications

Below are the publications that are part of the thesis.

- Chapter 3
  Rishabh Kaushal, Vasundhara Ghose, and Ponnurangam Kumaraguru *Methods for user profiling across social networks.* Accepted at the 12th IEEE International Conference on Social Computing (SocialCom), 2019.

- Chapter 4
  Rishabh Kaushal, Shubham Gupta, and Ponnurangam Kumaraguru *Investigation of biases in identity linkage datasets.* Accepted at the 35th ACM/SIGAPP Symposium on Applied Computing, SAC, 2020.

- Chapter 5
  Rishabh Kaushal, Shubham Singh, and Ponnurangam Kumaraguru *NeXLink: Node embedding frame-work for cross-network linkages across social networks.* Accepted at the International Conference On Network Science (NetSci-X), 2020.

- Chapter 6
  Rishabh Kaushal, Srishti Chandok, Paridhi Jain, Prateek Dewan, Nalin Gupta, and Ponnurangam Kumaraguru *Nudging nemo: Helping users control linkability across social networks.* Accepted at the International Conference on Social Informatics (SocInfo), 2017.

- Chapter 7
  Rishabh Kaushal, Chetna Sharma, and Ponnurangam Kumaraguru *Detection of misbehaviors in clone identities on online social networks.* Accepted at the 7th International Conference On Mining Intelligence and Knowledge Engineering (MIKE), 2019.

Other Publications: Below are the publications done during Ph.D. life which are not part of this thesis document.

- **Rishabh Kaushal**, Srishti Saha, Payal Bajaj, and Ponnurgam Kumaraguru *KidsTube: Detection, Characterization and Analysis of Child Unsafe Content and Promoters on YouTube.* Accepted at the 14th Annual Conference on Privacy, Security and Trust (PST), 2016.

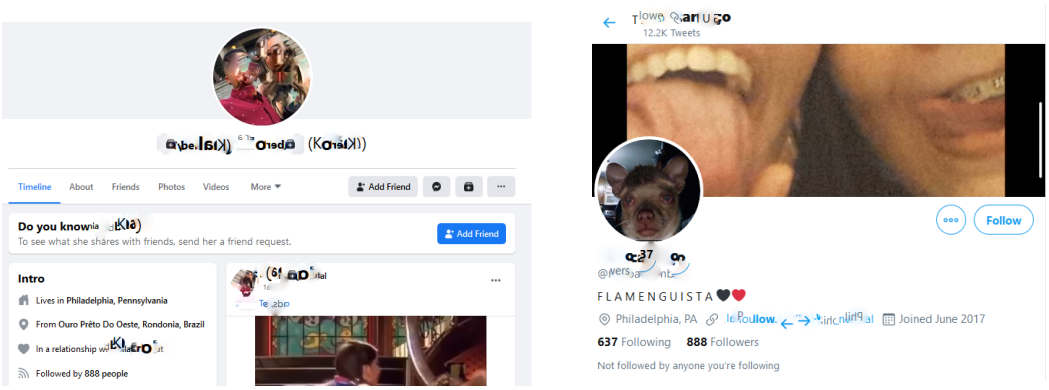
- Anjali Verma, Ashima Wadhwa, Navya Singh, Shivangi Beniwal, **Rishabh Kaushal**, and Ponnurangam Kumaraguru *Followee Management: Helping users follow the right users on Online Social Media.* Accepted at IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2018.

- Shubham Singh, **Rishabh Kaushal**, Arun Balaji Buduru, and Ponnurangam Kumaraguru *KidsGUARD: Fine Grained Approach for Child Unsafe Video Representation and Detection.* Accepted at the 34th ACM/SIGAPP Symposium On Applied Computing (SAC), 2019.

Chapter 2

Related Work and Background

The problem of User Identity Linkage (UIL) is known in literature by multiple names such as Social Identity Linkage [95], User Identity Resolution [8], Social Network Reconciliation [80], User Account Linkage Inference [140], Profile Linkage [180], Anchor Link prediction [79] and Detecting *me* edges [14]. Irrespective of the nomenclature, recall from Figure 1.2 that the goal is to connect identities belonging to the same user.



(a) Facebook Profile

(b) Twitter Profile

Figure 2.1: Illustration of user identities of the same user on Facebook and Twitter. Names and profile picture are blurred for privacy reasons.

To illustrate, let us assume that a user X has created accounts on *Twitter* and *Facebook*, as depicted in Figure 2.1 above. On Twitter, a picture of the dog is used as a profile picture, and there is a background picture, whereas, on Facebook, the user has shared an actual profile picture with a friend. The location attribute is similar across both identities. The challenge is to develop an approach that connects such identities to the same user. For illustration purposes, we have shown one such user, but it is required to perform this linkage for a large set of users in practice.

In order to understand the background and study the related works around the UIL problem, we collected all prior works by searching these names as the search keys on Google Scholar for the past years. After examining those prior works, we present a systematic study on the background of the problem user identity linkage from different perspectives as below.

- **Problem formulation**: We find that there are subtle variations in which the UIL problem has been formulated. Predominant formulation of the UIL problem in prior works [23, 42, 93, 122, 140, 175] is to decide whether the two given user identities on two different OSNs belong to the same person or not. However, other variations exist [22, 63, 112, 167, 192, 193] where the goal is to find *top-k* most likely matching identities in the target network corresponding to the given identity in the source network.

- **Data collection methods across OSNs**: In the UIL problem, the primary challenge is to collect ground truth user identities across multiple OSNs belonging to the same individual, referred to as *linked user identities*. We study various user behaviors namely aggregation of social identities [42, 93, 122, 180, 185], self-mention by users [23, 79, 88, 122, 133, 140, 175, 187, 189, 195], common email based sign-up across multiple OSN platforms [41], and snowball sampling [8, 96] that have been leveraged in past research in order to obtain linked user identities. We highlight some of the most commonly studied social networks used for data collection to solve the UIL problem, namely Twitter, Facebook, Instagram, and FourSquare.

- **Proposed approach adopted**: Conventionally, prior works address the UIL problem by looking at it as a machine learning problem and then developing supervised, semi-supervised, and unsupervised machine learning models. Past works have proposed novel ways to hand-craft features derived from profile [42, 88, 93, 122, 175], network (friends) [96, 191, 195] and content [3, 23, 41] posted by users across OSNs. However, with the recent advancements in graph representation learning [45, 179], we also found works [47, 94, 102, 124, 147, 148, 158] that automatically learn features as embedding vectors without the need to hand-craft the features. We perform a detailed study of both conventional approaches, and recent graph representation approaches to solve the UIL problem.

- **Implications and Applications**: In the last part, we discuss several problems in the area of social networks that would benefit from the solution of UIL problem. Problems of recommendation [114, 118, 119], link prediction [128, 183, 187], and many more can be more effectively solved once a comprehensive user behavior is obtained through the user's linked identities. We also discuss privacy implications [31, 35, 156] owing to the linkage of user identities and biases in identity linkage datasets.

2.1 Problem Formulations and Evaluation

In this section, we present two key formulations of the UIL problem and their evaluation approaches.

- Identity Linker: Learning an identity linkage function that *predicts* whether two given user identities on different OSNs belong to same or different user.
- Identity Extractor: Given a single user identity on an OSN, computing a function that finds *top-k* most probable identities corresponding to input identity on other OSNs.

2.1.1 Identity Linkage

The most commonly explored problem formulation in prior works [23,42,93,122,140,175] is to learn an identity linkage function that *predicts* or *classifies* whether two given user identities belong to the same individual or not. In this formulation, we model the function as a conventional machine learning-based binary classifier, which takes features related to user identities as input. We derive these features from user profile attributes, user content posting (and engagement with content), and network (friends) maintained by the user. More formally, we define the problem as follows.

Definition 2.1.1 *Given two user identities I_a and I_b on OSNs a and b, respectively, the goal is to learn a function F, which predicts whether I_a and I_b belong to the same individual or not.*

$$F(I_a, I_b) = \begin{cases} 1, & \text{if } I_a \textit{ and } I_b \textit{ belong to the same user.} \\ 0, & \text{otherwise} \end{cases} \tag{2.1}$$

We learn the function in two ways. The *first* approach is to create handcrafted features derived from the user's profile, content, and network. These features are then fed as input to the machine learning algorithms, as we explain later in Section 2.3. The *second* approach is to learn user identity representation (as we discuss in Section 2.4) in the form of an embedding vector and then apply machine learning algorithms on the learned embeddings. Given that we cast the problem as a binary classification problem, the standard evaluation metrics namely Precision (P), Recall (R), F1-score, True Positive Rate (TPRs), and False Positive Rate (FPRs) are employed. In the context of user identity linkage, we follow the evaluation approach as below.

1. We consider all possible identity pairs $< I_a, I_b >$ comprising of identities belonging to two social networks a and b as part of the input dataset D.
2. Each identity pair $< I_a, I_b >$ has a *label* associated with it, whose value is binary, either 1 or 0, indicating whether two identities I_a and I_b on OSNs a and b, belong to the same or different individuals, respectively.

3. We split the dataset D into training and test datasets. We use the label as supervisory information for learning of the function F. Evaluation is done based on standard metrics, as discussed in Table 2.1.

Table 2.1: Explanation of evaluation metrics in the context of the UIL problem.

Evaluation Metric	**Interpretation in context of UIL problem**
True Positive (TP)	User identities I_a and I_b belong to the same person and the learned function F also predicts the same person.
True Negative (TN)	User identities I_a and I_b do not belong to the same person and the learned function F also predicts different person.
False positive (FP)	User identities I_a and I_b do not belong to the same person but the learned function F says they belong to the same person.
False negative (FN)	User identities I_a and I_b belong to the same person but the learned function F says they do not belong to the same person.

4. Consequently, we redefine the standard classification metrics as below.

   - Precision (P): It is defined as the proportion of times the learned function F correctly predicts the input user identity pairs I_a and I_b to belong to the same individual.
   - Recall (R): It is defined as the proportion of user identity pairs I_a and I_b that belong to the same individual that the learned function F can retrieve out of total identity pairs belonging to the same person.

We describe prior works that have formulated user identity linkage as classification problem. Perito et. al. [122] modelled the UIL problem as binary classifier using features derived from username attribute only. Username pairs belonging to same person and different person were mixed with each other. Precision and recall were the metrics used to measure the performance of the username derived features fed into binary classifier. Zafarani et al. [175] looked at the problem of connecting user identities across social networking sites as classification problem. They paired usernames and derived features out of them to build a binary classifier with accuracy being the performance metric. Zafarani et al. [175] took username derived features for a given username and prior-username into consideration. They reported performance of logistic regression classifier with 92.72% accuracy in predicting correct username pairs belonging to same individual using top-10 features. Goga et al. [42] used the similarity scores between profile attributes of user identities on two social networks as the classification features. Liu et al. [93] and Chen et al. [23] considered the problem of linking user identities as binary classification problem in which two usernames are provided as input. Liu

et al. [93] used the n-gram username probabilities to build a classification model to decide whether username pair belong to same user or not, and therefore, they used accuracy as a metric for evaluation. Their proposed approach based on n-gram username probability achieves an accuracy of over 90%. Shen et al. [140] proposed User Accounts Linkage Inference (UALI) framework whose goal was to learn function which has binary outcome 1 and -1 depending upon whether two input user identities belong to same person or not, respectively. Chen et al. [23] used precision and recall as evaluation metrics. Number of correctly linked user identity pair among the total identity pair returned as result was defined as precision. Recall was defined as number of correct user identity pairs detected from among the total correctly linked identity pairs. Goga et al. [41] considered cosine distance to measure similarity of user's location profiles represented using Term Frequency and Inverse Document Frequency (TF-IDF) based vectors. True positive rate (TPR) and false positive rate (FPR) were computed based on different similarity scores. In addition, they also considered accuracy by considering the problem as classification problem using logistic regression classifier. Almishari et al. [3] approached the problem of linkability using the stylometric features of user's content within same social network. Tweets from same user were split into two groups, one referred as Identified Record (IR) and other as Anonymous Record (AR). Classifiers were trained on IRs in which tweets along with associated labels identifying the users who posted the tweet are mentioned. ARs were used for the purpose of evaluation, the goal is to link each record in AR to one of the user. Classifier performance was measured in terms of Linkability Ratio (LR) which was computed as number of records in AR correctly associated within *top-n* candidates. With larger sizes of IRs and ARs, the linkability ratio increased. Shen et al. [140] used conventional classification algorithms namely Decision Trees, Naive Bayes, SVM and Adaboost, as representing the function F in the learning process. focus on three social networks namely Google+, Twitter and Foursquare. Area under the curve (AUC) score was the evaluation metric used, higher the AUC score the better was the performance. Zhang et al. [191] used the term, *network reconciliation problem*, for linking user identities across social networks and represented it as classification problem, thereby using F1-score, precision and recall as evaluation metrics. Zhang et al. [189] also presented the user linkability problem as binary classification problem. Their proposed COSNET framework performed better than conventional classifiers like SVM by a margin of 10-30% in terms of F1-score. In addition, the impact of user linkability was also studied on the problem of finding expert. Linked user information was augmented with existing information about the user to make better decisions in identification of experts. For instance, knowledge of both ArnetMiner, an academic social network and LinkedIn, a professional network, is combined together to find experts in ArnetMiner.

2.1.2 Linked Identity Extractor

The other way for formulating the problem of user identity linkage (UIL) is to learn a *ranking function* which given a single user identity on one social network (source), *orders* the identities on

another social network (target) such that correct linked identity appears among the *top-k* identities extracted from the target network. In this formulation, prior works [22, 63, 112, 167, 192, 193] model the ranking function as a conventional ranked retrieval problem from the field of information retrieval (or extraction). Like, the binary classifier function, we compute this ranked retrieval function using the features derived from profile, network and content of user identities, details are presented in Section 2.3. More formally, we define the problem as follows.

Definition 2.1.2 *Given a user identity I_a on source OSN_a , the goal is to learn a function F_{rank} that finds top-k user identities $< I_b^1, I_b^2, ..., I_b^k >$, one out of which is likely to belong to the same individual whose identity I_a on OSN_a is already known.*

Alternatively, in recent times, we learn embedding vectors that represents user identity and we compare these embeddings to obtain a rank score which is used to rank identities, details are presented in Section 2.4.

Table 2.2: Explanation of evaluation metric in the context of user identity linkage

Evaluation Metric	**Interpretation in the context of UIL problem**
Success/Hit at Rank k (S@k)	The proportion of times that correct linked identity I_b is present among the top-k identities that we retrieve.
Mean Reciprocal Rank (MRR)	The average rank at which the linked identity I_b occurs in the top-k identities that we retrieve.

Given that we cast the UIL problem as a *ranked retrieval problem*, we adopt the following evaluation approach.

1. We consider all possible identity pairs $< I_a, I_b >$ comprising of identities belonging to the two social networks a and b to be part of input dataset D.
2. For each user identity I_a in linked identity pair $< I_a, I_b >$, using different ranking functions, we find an ordered list of identities $< I_b^1, I_b^2, ..., I_b^k >$.
3. Subsequently, we perform evaluation on the basis of metrics discussed in Table 2.2.

We discuss few prior works which formulate the user identity linkage (UIL) problem in terms of a ranking function. Iofciu et al. [63] relied upon the tags that users had placed in their profiles across social networks. Given a user's identity on a source social network, they ranked the user identities on the target social network with the hope that the linked identity appeared at the top of the ranked list. Mu et al. [112] presented an approach to project users across social networks into a latent user space such that those users exhibiting similar characteristics are closer in this latent

space. They considered the UIL problem as an extraction problem, and for evaluation, they used *hit(x)* to represent the position at which correctly linked user identity is present among the *top-k* identities returned from target social network. Zhou et al. [192] proposed a deep learning based approach, referred to as *DeepLink*, which learns node representations based on network structures. For evaluations, they employed metrics, namely mean average precision (MAP) and precision at top-k. Xie et al. [167] leveraged the concept of factoid to create user embeddings where each user is represented as a triple comprising of user identity, object, and predicate. Evaluation metrics used were HitRate@K and mean reciprocal rank (MRR). Chen et al. [22] proposed INformation FUsion and Neighborhood Enhancement(INFUNE), a novel framework for the fusion of information and enhancement of neighborhood. They used heterogeneous information describing users to generate user node embeddings through encoder-decoder models. They employed hit-precision as the evaluation metric to find linked user identity in top-k candidates. Zhou et al. [193] presented *TransLink*, an approach based on translation-modeling, which creates user embeddings based on user behaviors modeled as interaction meta-paths. For evaluation, they used mean rank, which indicates the average position at which linked user identity was found in the target social network.

Having discussed the two key formulations for the UIL problem, we discuss methods for collecting linked user identities in Section 2.2.

2.2 Data Collection

We recall that Online Social Networks (OSNs) offers a variety of services to their users, and therefore, users join more than one OSN platform to avail these services, which leads to the problem of User Identity Linkage (UIL). In order to solve the UIL problem, the collection of ground truth user identities belonging to the same person across different OSNs, is an essential first step. We refer to these identities as *linked user identities*. In this section, we present several methods used to obtain linked user identities and then find the different social networks used by prior works from where user data was collected.

2.2.1 Methods for Linked User Identities Collection

We organize and present methods to collect linked user identities. In Fig 2.2, we depict a generic framework for data collection, data integration, and data extraction & indexing. In the first step, we identify a data source (a social network in our case), and in the second step, we select user behavior based on which a data collection method get decided. Once the linked user identities are collected, we integrate and store them in a common data pool, referred to as Linked Identity Data Store (LIDS). Next, we present different user behaviors based on which data collection methods are designed.

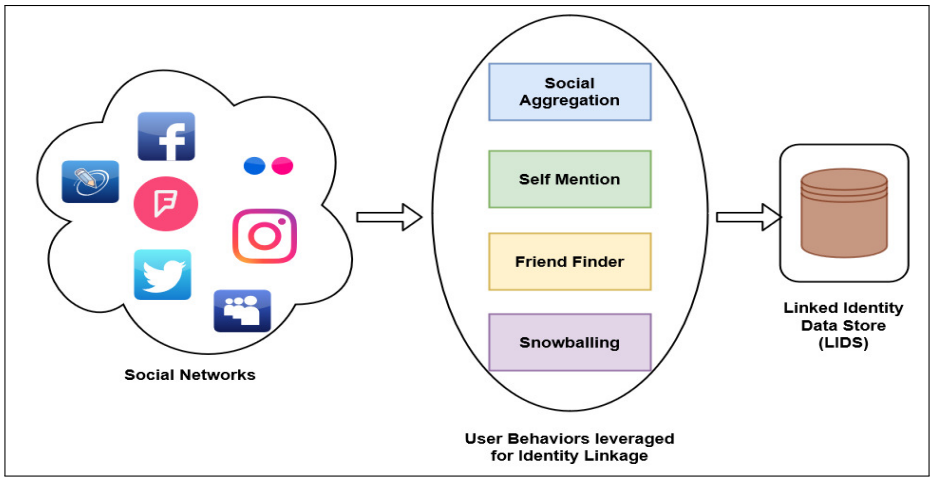


Figure 2.2: Framework to collect linked user identities from different social networks and their storage.

Aggregation of Social Identities

There are several online platforms where users aggregate their social identities (accounts) together at a common place. We refer to this behavioral phenomenon as *social aggregation* and these online platforms as social aggregators. Many prior works [42, 93, 122, 180, 185] leverage from this user behavior to obtain linked user identities across social networks. One such social aggregator is *about.me*[1], where users mention their social identities on Twitter, Facebook, LinkedIn, Flickr, Pinterest, YouTube, Tumblr and so on. The social aggregators provide options to search user profiles using different criteria like users' interests, which are exploited in prior works to crawl users' data. Next, we discuss some of these prior works. Perito et al. [122] performed large scale crawling on public Google profiles and eBay accounts to obtain 3.5 million and 6.5 million usernames. Liu et al. [93] crawled 75,472 public profiles on a social media aggregator site called about.me where users mention details of their identities on at least two social media sites. Total of 15 different social media sites were mentioned by these users, with each user mentioning on average 3.92 social media sites on their About.me profile page. Besides, they also conducted a survey comprising of 153 participants and found that around 82% of them participated in 1-4 online social media sites. One of the contributions of their work was to find the rareness or commonness of usernames, for which they collected usernames by searching through 69 million question-answer threads in Yahoo! answers. From these, they sampled 299,716 usernames mentioned by 673,037 unique users. Goga et al. [42] crawled 3 million Google+ accounts to find ground truth and leveraged the fact that users on Google+ can mention their social media accounts on other websites. Besides above, they also obtained ground truth of 19,000 user pairs on Flickr and Twitter using friend finder feature based on emails. Zhang et al. [185] obtained ground truth by leveraging the fact that users on

[1]About.me: https://about.me/

Question-Answer social networking sites mention details of their other accounts on their home pages on these sites. Around 10,000 users from three sites, namely Stack Overflow, Super User, and Programmer Q& A were obtained, out of which around 20-30% users match pair-wise. Zhang et al. [180] sampled 152,294 Twitter profiles from the tweets posted by users and parse 154,379 profiles from LinkedIn. For ground truth, they looked at Google+ profiles of users and find 9,750 user identities that belong to both Twitter and LinkedIn.

Self Mention by Users

At the time of account creation and later as well, users have the option to configure their profile on OSN platforms. In the profile settings, there are options to provide their social identities on other OSN platforms. We refer to this user behavior to self-mention their identities on other OSN platforms, as *self-mention* or *self-disclosure*. Many prior works [23, 79, 88, 122, 133, 140, 175, 187, 189, 195] leverage this user behavior to collect user identities of the same person across different OSN platforms. Zafarani et al. [175] collected 100,179 username and prior usernames from many sources, namely blogs, forums, and social networking sites. Their work was unique in the sense that they obtained these usernames pairs from 32 different sites, the maximum coverage any work has done so far. Li et al. [88] leveraged the unique numeric user ID of users on location based social network, Foursquare. On their profile page on Foursquare, some users mention URLs of their Twitter and Facebook profiles. Out of the 1.3 million identities crawled on Foursquare, they could get only 597,822 profiles that were public and available. Among these, 288,480 profiles mentioning Facebook identity, 102,315 profiles mention Twitter identity and 67,826 profiles mentioned both Facebook and Twitter identity. Chen et al. [23] leveraged trajectory and check-in data in three real-world datasets. The first dataset comprised walk trajectories of users capturing their outdoor movements like cycling, shopping, driving and site-seeing. This data comprised of 182 user pairs containing 14,337 walk trajectories with 2,190,957 locations and 5,475 car trajectories with 925,380 locations. The second dataset comprised of 89 user pairs from Twitter-Foursquare containing 3,924 check-ins on Foursquare and 35,384 check-ins on Twitter. The third dataset consisted of 908 pair of users from Instagram and Twitter, comprising of 267,029 check-ins in Instagram and 357,949 check-ins on Twitter. Shen et al. [140] focussed on three social networks, namely Google+, Twitter, and Foursquare. They collected data using the APIs of these networks and also use crawling to collect more details of users like their neighborhood information. They used common screen names across Twitter and Google+ to find linked Twitter - Google+ user pairs. Some users mentioned details of their Twitter and Google+ account on their Foursquare profiles, which they used to construct linked Twitter - Foursquare and Google+ - Foursquare user identity pairs. Zhang et al. [189] considered five social networks, namely Twitter, LiveJournal, Flickr, Last.fm, and MySpace. They obtained ground truth linked identity dataset from the prior work of Perito et al. [122]. In addition to social networks, they also used datasets comprising of academic data,

namely Arnet-Miner, LinkedIn, and VideoLectures. Arnet-Miner is a platform where users mention details of their other networks (like LinkedIn), which helped in ground truth data for these academic social networking platforms. Zhou et al. [195] evaluated their FRUI (Friendship Relationship Based User Identification) algorithm on both synthetic and real-world datasets. For synthetic datasets, they used random networks [36], small-world networks [166] and preferential attachment model based networks [7], with each network comprising of 10,000 nodes. For real networks, they captured data from the Sina Microblog search page and use OpenAPI to collect RenRen dataset. Kong et al. [79] used the self mention information of Twitter identities on the Foursquare profile of users to link their identities on Foursquare with Twitter. In total, they obtained 500 ground truth matching users on both Foursquare and Twitter. Zhang et al. [187] crawled two social networks Foursquare and Twitter, around November 2012. They crawled 5,392 users from Foursquare to obtain 48,756 tips and 38,921 locations. From Twitter, they crawled 5,223 users and retrieve 9,490,707 tweets. Sajadmanesh et al. [133] used 3456 Foursquare users and 5223 Twitter users as the two social networks. Ground truth comprised of 3282 out of which 1900 users joined the target network after joining the source network.

Common Email based registration across OSN platforms

Users register themselves across multiple OSN platforms using their same email address. Let us assume that there are two users X and Y, who communicate with each other over their respective emails and thus have each other's email in their email contact list. Now, let's say X joins a social network A using her email address, which has Y's email in her email contact list. Assuming that Y has already joined this social network A using her email, then Y's identity in social network A is *recommended as a friend* either implicitly or explicitly to X who has recently joined. We refer to this feature offered by many social networks as *friend-finder*, which have been used by prior works to collect linked user identities. Goga et al. [41] leveraged the mechanism of *friend-finder* in social networking sites. An extensive collection of 10 million emails were used to link accounts belonging to these emails on three social media sites namely Twitter, Flickr and Yelp. Number of linked users in Twitter-Flickr, Twitter-Yelp and Flickr-Yelp are 13,629 , 1,889 and 1,199 , respectively. Subsequently, they reorganized this data across five localities in US namely Los Angeles, New York, Chicago, San Francisco and San Diego). To get metadata associated with tweets and photos, they used Twitter API and Flickr API, respectively. In the case of Yelp, profile pages were crawled and parsed to extract relevant information.

Snowball Sampling (SS)

In the context of a collection of linked identities, snowball sampling would refer to the process where we increase the linked identities collection by searching in the neighborhood of known linked

identities (referred to as seed pairs). Bartunov et al. [8] started with a seed of 16 users on Twitter and Facebook, and used a snowball sampling to collect 398 and 977 users on these two social networks, respectively. For Twitter, they used mutual following as an equivalent of friendship relation in Facebook. Liu et al. [96] accessed user behavior data on Douban using it's API which is Chinese social networking site allowing users to create content related to books, movies, music, and local events in cities. A random set of 20 users were selected and their network was crawled using breadth first search approach to increase the number of users to 50,000.

Miscellaneous

Besides the above methods, few prior works have adopted data collection methods that do not fall under any of the methods mentioned above, therefore, we discuss them in this miscellaneous category. Almishari et al. [3] extracted two small subsets from the set of tweets collected by a prior study done by Yang et al. [169] across six month period in 2009. The first subset comprised of 8,262 users who have tweeted more than 2,000 tweets and the second subset contains tweets (around 300 - 400 per user) from 10,000 randomly selected users. They divided each user's tweets into two sets namely Identified Record (IR) and Anonymous Record (AR). Further, they used stylometric features to *link* user's tweets across IR and AR. Zhou et al. [195] evaluated their Friendship Relationship Based User Identification (FRUI) algorithm on both real-world and synthetic datasets. For synthetic datasets, they used random networks [36], small world networks [166] and preferential attachment model based networks [7], with each network comprising of 10,000 nodes. For real networks, they captured data from the Sina Microblog search page and use OpenAPI to collect the RenRen dataset. Zhang et al. [191] used the Facebook dataset provided by Viswanath et al. [154] comprising of 63,731 nodes and 817,090 edges and synthetically generate two sub-graphs. Nie et al. [116] identified the core interests of users based on tweets from 1,000 random Twitter users over 12 months period. Further, for evaluating linking of profiles across social networks, they targeted 1,213 user pairs from Twitter and BlogCatalog, a social site that allows users to join communities, thereby indicating user interests. Zhang et al. [190] collected details of 20,448 and 40,618 users on two popular Chinese social networks namely Sina Weibo (similar to Twitter) and Renren (similar to Facebook), respectively. For ground truth, they manually linked user identities from these two social networks.

To summarize, Table 2.3 provides the distribution of prior works among the various data collection methods discussed in this section. Most of the works have used social aggregation or self-disclosure as their data collection methods. Prior work rarely use the friend finder method because of the dependence on the availability of emails.

Table 2.3: Distribution of prior works among the data collection approaches for collecting linked identities.

Name of Method	Prior Works
Self-Disclosure (SD)	[175], [88], [23], [140], [189], [122], [195], [79], [187]
Miscellaneous	[3], [169], [195], [191], [154], [116], [190]
Social Aggregator (SA)	[122], [93], [42], [185], [180]
Snowball Sampling (SS)	[8], [96]
Friend Finder Feature (FFF)	[41]

2.2.2 Social Network Diversity

Prior works cover several social networks. In this section, we present the distribution of social networks covered by researchers to solve the problem of user identity linkage in the past. Table 2.4 provides the list of social networks, it may be noted that each prior work appears two or more times because each work collects user identities from two or more OSN platforms. From Table 2.4, we observe that most of the prior works use Twitter as the social media platform because data on Twitter is public by default and it provides excellent support for Application Programming Interface (API), which is a collection of pre-defined functions used to obtain Twitter data through computer programs. After Twitter, we find that many prior works collect data from location-based social network Foursquare and image-based social network Flickr. Following them, we observe that social networks, namely Google+, Facebook, MySpace, and LiveJournal, are the platforms for data collection. While Facebook is the most widely used social network, the reason for the low adoption of Facebook in the research community is because the Facebook graph API is restrictive owing to the nature of private content, which is mostly present on Facebook. Prior works sparingly use remaining social networks.

We provide below a few indicative prior works along with the details of social networks being used by them. Perito et al. [122] conducted studies on using only usernames. They investigated large lists of usernames comprising of 3.5 million usernames obtained from public Google profiles, 6.5 million from eBay accounts. They used the information expressed on Google profiles to derive linked user identities. Zafarani et al. [175] did not restrict themselves to only social networking sites. They obtained username pairs from various other sources like web blogs and forums. In total, they collected usernames from 32 online sites. Li et al. [88] leveraged the incremental numeric user IDs on Foursquare to collect ground truth. From the Foursquare profile pages of users, they gathered self-disclosed identities of users on two other social media sites, namely Facebook and Twitter. Liu et al. [93] crawled 75,472 public profiles on About.me and collect a total of 15 different social media sites mentioned by these users. Goga et al. [41] considered data from three social media sites, namely Yelp, Twitter, and Flickr offering various content sharing services to users in terms of service reviews, micro-blogs, and photo sharing, respectively. Chen et al. [23] obtained datasets on Instagram-Twitter and Foursquare-Twitter from prior work of Riederer et

Table 2.4: Distribution of social networks from where user identities are collected by prior works.

Social Network	**Prior Works**
Twitter	[8] (2012), [42] (2013), [79] (2013), [41] (2013), [3] (2014), [140] (2014), [10] (2014), [187] (2014), [180] (2014), [189] (2015), [186] (2016), [133] (2016), [88] (2017), [23] (2017),
Foursquare	[79] (2013), [140] (2014), [187] (2014), [186] (2016), [133] (2016), [23] (2017), [88] (2017)
Flickr	[63] (2011), [41] (2013), [42] (2013), [10] (2014), [189] (2015)
Google+	[122] (2011), [42] (2013), [140] (2014)
Facebook	[8] (2012), [42] (2013), [88] (2017)
MySpace	[42] (2013), [189] (2015)
LiveJournal	[10] (2014), [189] (2015)
About.me	[93] (2013)
Blogs	[175] (2013)
Delicious	[63] (2011)
Douban	[96] (2017)
Instagram	[23] (2017)
Last.fm	[189] (2015)
LinkedIn	[180] (2014)
Stack Overflow	[185] (2015)
StumbleUpon	[63] (2011)
Super User	[185] (2015)
YouTube	[10] (2014)
Yelp	[41] (2013)

al. [130] and pruned the data to only those data instances which contain sufficient trajectories. Besides these, they also evaluated their approach to walk and car trajectories data from Beijing's GeoLife project.[2] Almishari et al. [3] looked at the problem of linking content posted by users within single social network, namely Twitter. They divided the tweets posted by the user into two parts and recast the linkability problem as detecting the same user's posts across these two parts. Shen et al. [140] focussed on three social networks namely Google+, Twitter, and Foursquare. Zhang et al. [189] worked on data from five social networks, namely Twitter, LiveJournal, Flickr, Last.fm, and MySpace. Additionally, they also used datasets comprising of academic content, namely Arnet-Miner, LinkedIn, and VideoLectures. Iofciu et al. [63] linked users across three social networks, namely Flickr, Delicious, and StumbleUpon. While Flickr is an image sharing platform, the remaining two help users organize their publicly available web documents. Kong et al. [79] collected user data from Foursquare, and Twitter. They employed breadth-first search strategy using the 7,504 tips (location updates) information as a seed to obtain 500 users on

[2]https://www.microsoft.com/en-us/research/people/yuzheng

Foursquare. Further, corresponding to these users, another 500 users on Twitter were collected with 741,529 tweets. Bartunov et al. [8] collected 398 and 977 user identities on Twitter and Facebook, respectively, starting with 16 seed pairs of nodes. Goga et al. [42] studied five popular social networks namely Facebook, Twitter, Flickr, Google+, and MySpace. Bennacer et al. [10] worked on four social networks YouTube, Flickr, Twitter, and LiveJournal. They extended the dataset provided by Buccafurri et. al. [14] by filling the missing attribute information and adding new friend connections using the APIs of these networks. Zhang et al. [187] evaluated their Multi-Network Link Identifier (MLI) framework on Foursquare and Twitter social networks, comprising of around 5,000 users from each of the network. Zhang et al. [185] focussed on linking users across Question-Answer based social networks namely Stack Overflow, Super User and Programmers Q&A. Zhang et al. [186] used Foursquare and Twitter as the two social networks with both users and locations co-aligned as the ground truth. Sajadmanesh et al. [133] also used Foursquare and Twitter as the two social networks. Zhang et al. [180] performed profile linkage using cost-sensitive features on Twitter and LinkedIn social networks. Liu et al. [96] used Douban, which is a Chinese social network that provides facility to user to create content related to films, music, books, and events in various cities.

2.3 Machine Learning Approach

In this section, we discuss the machine learning approach to solve the UIL problem. As per this approach, depicted in Fig 2.3, we leverage profile, content, and network information of the users to create features. We next describe these features.

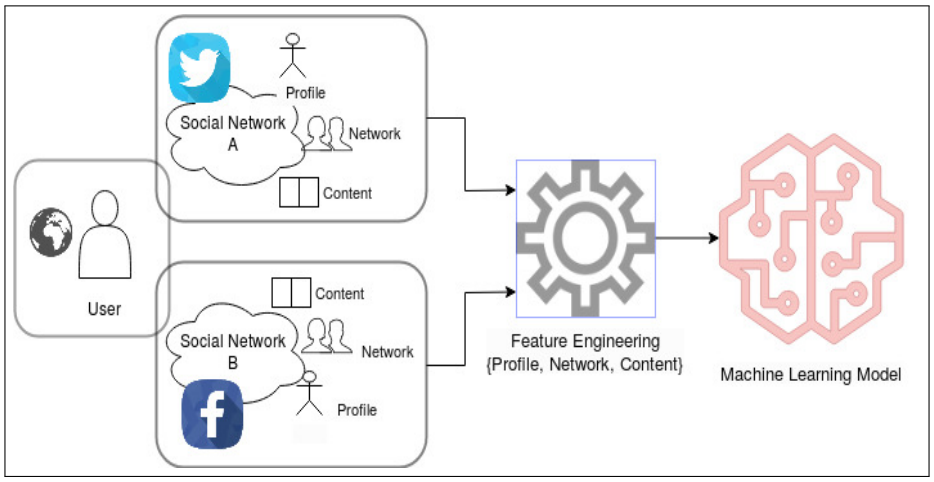


Figure 2.3: Broad framework for solving user identity linkage problem. A user identity has three dimensions namely profile, content, and network. Features are derived from one or more of these dimensions and are passed as input into machine learning based classification model.

2.3.1 Profile Features

Profile features comprise of user's basic information like username, display name, location, and profile picture. OSNs have different options and interfaces with varying degrees of details to represent user profile features. Given that access to user's content and network (friends) has been dwindling due to privacy considerations, there are works in the past that have restricted themselves to the use of only profile features.

One of the earliest works by Perito et al. [122] proposed to connect user identities only based on usernames. They applied the concept of *information surprisal*, which quantifies the amount of information that the outcome of an experiment conveys. For random variable X and x as one of the outcome, the information surprisal is defined as $I(x) = -logP(x)$, which suggests that low probability gives a higher surprisal. They found that usernames alone express much information quantified by information surprisal. Besides, they argued that the probability of two usernames belonging to the same person depends on the shared information conveyed by these usernames and likelihood of user changing username from one form to another. They proposed three approaches to compute this likelihood. The first approach modeled it as a Markov-Chain process, in which the goal was to predict the next character of the username. The second approach used TF-IDF, where they considered characters as terms and all possible substrings of given usernames as documents. In the third approach, they used string-only similarity metric, namely levenshtein distance to measure the similarity between two strings.

Table 2.5: Features derived from profile attributes in prior works.

Profile Attributes	**Feature Description**	**Prior Work**
Username	Information surprisal, which quantifies the amount of information that an experiment conveys.	Perito et al. [122]
	Patterns used in creating usernames like typing styles and language influences.	Zafarani et al. [175]
	Redundant information like same characters, similar distribution of alphabets.	Li et al. [88]
	Rare-ness or common-ness in usernames, using n-gram probability.	Liu et al. [93]
Profile picture, Location, Username Name.	Correlation between these profile attributes, and chain of correlation among five OSNs, if high correlation is not found in an OSN pair.	Goga et al. [41]
User Tags	TF-IDF vectors of these tags assigned by users.	Iofciu et al. [63]

Zafarani et al. [175] proposed a framework called MOBIUS (modeling behavior for identifying users across sites) for connecting user identities across social media sites. The framework comprised of three steps. In the first step, users were identified by their unique behaviors, thereby resulting in

redundancies across social media sites. In the second step, they generated features that are based on these redundancies. Finally, in the third step, the features were fed into machine learning classifiers. MOBIUS used the most basic information, that is, username as the user attribute to measure user behaviors. It created an extensive set of features based on patterns due to exogenous factors, human limitations, and endogenous factors. While creating usernames, humans were constrained by knowledge limitation, memory & time limitations. The exogenous factors affecting users' decision to create usernames were typing and language patterns adopted by humans. They extracted a total of 414 features by leveraging these factors, out of which top-10 features were finally considered after performing feature importance. Li et al. [88] investigated the redundant information associated with usernames of users across social networks. They captured the redundant information in terms of length of username, similar characters in the username, and similarity in the distribution of letters in username. As per their findings, around 45% of users kept the same usernames across social networks. Goga et al. [42] found a correlation between readily available attributes, namely username, profile pic, location, and real name. They obtained classification features from comparisons of these attributes on five different social networks. If two accounts belonging to the same user do not exhibit a high correlation for a particular pair of the social network, then the chain of correlation is explored to link user accounts using correlation of attributes with third social network. Liu et al. [93] looked at the problem from the perspective of *alias-disambiguation* which tells whether two same usernames belong to the same person or not. They solved the problem by proposing a methodology for automatic labeling of usernames. They hypothesized that usernames which are rare would belong to the same individual whereas username which is common would belong to different person. They computed the rareness or common-ness of usernames using the n-gram username probability. To this end, they segmented the given username into words and then find the probability of the words in the given corpora. Logistic regression function was applied to the n-gram username probability to find whether two given usernames belong to the same person or not. Furthermore, they claimed that this model outperforms the model which is using features derived from user meta-data like avatar, location and user's post based features. Iofciu et al. [63] leveraged the user assigned tags to the user profile on different social networks. They used TF-IDF based vectorization to consider each user's profile as a vector of tags associated with the profile. Cosine distance was the metric used to compare two vectors representing two user profiles.

2.3.2 Content Features

In this section, we discuss prior works that derive features from the content posted by users on various OSNs. Goga et al. [41] studied the content posted by users across different social networks and propose a solution using which adversaries can match user accounts belonging to the same person. They investigated three characteristic features associated with posted content, which include the timestamp of post, the writing style of the user, and the geo-location with the post. For

locations, they used the zip code of users. Histogram representing the frequency of visits of users to a particular location is used as a *location profile* of the user. TF-IDF weights on zip codes for a user are used to construct location features for the user. For the timestamp of the post, authors exploited the automated cross-posting behavior of users across social networks. Posts made within a short time period, obtained from ground truth, were considered coming from the same users. Lastly, they considered the content of the post made by users across social media sites. Language models were constructed based on the histograms of unigrams occurring in the user posts. Features derived from posts, timestamps and locations were passed as input to binary logistic regression classifier. They found that location and timestamp play a more critical role in identifying users than the content of posts.

Table 2.6: Features derived from content attribute (user posts) in prior works.

Content	**Feature Description**	**Prior Work**
User Posts	Timestamp, writing style and geo-location of posted content.	Goga et al. [41]
	Spatio-temporal features considered as continuous time and space variables.	Chen et al. [23]
	Unigrams and bigrams extracted from alphabets used in user posts.	Almishari et al. [3]

Chen et al. [23] proposed a novel STUL (spatio-temporal user linkage) model, which extracted the spatial and temporal features of users to link user identities across social networks. They considered both time and space as continuous variables. They used an extension of density-based clustering to obtain spatial features of users, which were captured as *stay regions* as places where user has stayed. To extract temporal features of users, they used Gaussian Mixture Model (GMM), which contains global and local time distributions. Features from space and time are assigned weights based on the TF-IDF approach. Two types of user data were monitored namely trajectory of the user and the check-in data from the user. Almishari et al. [3] showed that users maintaining multiple accounts on Twitter can be linked to the same person in the presence of large number of Twitter users provided they are actively posting tweets. Two categories of text features were extracted, namely unigrams comprising of all english letters and bigrams consisting of all possible two-letters found in tweets. These features were used in Naive Bayes classifier to decide the user who has posted the tweet.

2.3.3 Network Features

One of the fundamental principles of social networking is the concept of *homophily*, which implies similar users connect with each other. User's network information is an essential feature for linking user identities. Zhou et al. [195] proposed FRUI (Friendship Relationship Based User Identification)

algorithm, which used the fact that identical users set up common friendship structures in different social networks. Given two user identities I_a and I_b from two social networks a and b as input, the algorithm aims to find the match degree $M_{i,j}$ which was defined in terms of common neighborhood. Zhang et al. [191] observed that users have different tie strength across social networks with their friends, which they referred to as heterogeneous relationships. The degree of interaction among two users decided the tie strength. They proposed network reconciliation algorithm (*NR-GL*) that leverages this heterogeneous relationship among users, into a unified framework, *UniRank*, comprising of local and global features. Proposed algorithm started by exploring seed user pairs (similar user identities across social networks) and then for each such pair, used a breadth first strategy with local matching to find more such seed pairs. UMA leveraged the fact that social networks have few common users across them are called as *partially aligned networks*, and such users are referred to as *anchor nodes*. Liu et al. [96] approached the problem of linking users across different social networks by proposing a model that measures the distance of users across social networks, referred to as the Adaptive User Distance Measurement (AUDM) model. Model casts the problem as a convex optimization problem, converts each social network into a common embedding space, leverages metric learning, and boosting to find the distance between users.

2.3.4 Profile and Network Features

We discuss prior works that derive features using both profile and network information. Shen et al. [140] focussed on raising awareness of the risks associated with linking user identities across social networks. In particular, they proposed a User Account Linkage Inference (UALI) framework, which helps in making users aware of the risks due to the linkage of user identities. Subsequently, they introduced a mechanism to enable users control the risks associated with identity leakage through their proposed framework, referred to as the Information Control Mechanism (ICM). The UALI framework used basic features obtained from profile (name, gender, location) and neighborhood (friends, followers, and followees). Zhang et al. [189] proposed a novel energy-based model, referred to as COnnecting heterogeneouss Social NETwork (COSNET) which incorporates local user matching based on the profile information of the user and network matching based on neighborhood information of the user. Besides, since the work focused on more than two social networks, they consider global consistency which states that if I_a, I_b and I_b, I_c are linked user identity pair on social networks a, b and b, c, respectively, then by transitivity, I_a, I_c is also linked pair across networks a, c. They obtained an objective function by combining local, network, and global consistency. Zhang et al. [180] proposed an approach to profile linkage that leverages cost-sensitive features, namely profile avatar and geocode using Google Maps API, besides the common friend information. Their approach made use of local features, namely username, language, profile description, and network popularity. Bartunov et al. [8] introduced a probabilistic approach based on conditional random fields, referred to as Joint Link-Attribute (JLA), to find user identities of single-user across social

networks. They used *scheme mapping* [85] to align two key user attributes, namely screen name and URLs provided by the user in their profiles of social networks. For comparing common network structures, they used the *dice coefficient*, which is the normalized form of common nodes directly connected to the given node pair. Zhang et al. [190] proposed a local expansion strategy based on the breadth first search to find user identities belonging to the same user. They used profile and network based features, namely username, home town, and friend network to expand the initial small seed linked users, referred to as known anchor links. Bennacer et al. [10] leveraged publicly available profile information along with topology of users' friend network to link user accounts across the social networks. The first step involved the selection of candidate pairs of users who are likely to belong to the same individual based on network topology. In the second step, they used public attributes to create matching rules to compare two user accounts. Zhang et al. [186] linked not just common users across social networks, but also common locations being referred across social networks. They proposed unsupervised concurrent alignment (UNICOAT), which leverages attribute and link information to recast the alignment problem as a joint optimization problem. Their work relied on the observation that users have common neighbors and profile attribute information across social networks, the quality of this common-ness is captured in the cost function.

2.3.5 Content and Network Features

Nie et al. [116] proposed a Dynamic Core Interest Mapping (DCIM) algorithm that builds upon limitations of human behaviors in social networks. As a consequence of human limitations, the core interests of users were limited. Moreover, the DCIM algorithm computed core interests of users and then used it to map user identities across social networks. Content posted by users, along with the structural connections shared by users with their friends, were jointly used in the algorithm. Zhang et al. [185] focused on multiple anonymized social network alignment problem in which an unsupervised approach which relied on transitive relation among user accounts across social networks. They referred their proposed approach as Unsupervised Multi-network Alignment (UMA) to align multiple networks in which users are anonymized to protect their identity. UMA leveraged the fact that social networks have few common users across them, referred to as *anchor nodes*. Question-Answer types of social networks were considered, and an edge between two users was considered if they both post on the same question. This edge information was used to cast a pairwise network alignment problem as optimization problem. Kong et al. [79] proposed a Multi-Network Anchoring (MNA) framework, which captures heterogeneous features of users across social networks. They derived the first set of features from the social connections of users across social networks. In particular, the notion of a common network (friend circle) was captured in three different metrics, namely common neighbors, Adamic/Adar measure, and Jaccard coefficient. They considered the content posted by users as weighted TF-IDF vectors. Additionally, they also considered the location and time of the

user posts as features derived from content. Zhang et al. [187] proposed a Multi-Network Link Identifier (MLI) framework, which was based on the creation of intra-network and inter-network social meta paths. The social network was modeled as a graph comprising of nodes of different kinds - users, posts, words appearing in posts, the time stamp of posts, and locations from where posts were made. Homogeneous meta paths captured the relationship between the same type of node, in this case, user-user relationships based on follow-followee relationships. Heterogeneous meta paths captured the relationship between dissimilar types of nodes, in this case, user-content relationships based on location, timestamp, and words appearing in a post. Mutual information based on information theory was used as the ranking metric to identify important meta paths. They used the features from these meta paths to build link prediction models. Sajadmanesh et al. [133] also used meta-path based approach, in particular, they proposed two types of meta-paths namely Connector and Recursive Meta-Paths (CRMP). Like Zhang et al. [187], they too created paths comprising of user nodes, user posts, words in the post, time and location of the posts. They constructed six different types of meta-paths based on user social connections (follower-followee relationship). Other types of meta-paths were based on the temporal, spatial, and textual similarity of posts made by users. Path count, in other words, a number of meta-paths for each node in the target network, was used as the feature for the classifier, which is SVM with a linear kernel.

2.3.6 Summary

To summarize, we organize prior works which use hand-crafted features as input into machine learning classifiers in Table 2.7.

Table 2.7: Prior works who have used hand-crafted features as inputs to machine learning classifiers for solving the User Identity Linkage (UIL) problem.

Source of Feature	**Prior Works**
profile	[122], [175], [88], [42], [93], [63]
content	[41], [23], [3]
profile and network	[140], [189], [180], [8], [85], [190], [10], [186]
content and network	[116], [185], [79], [187], [133], [187]
network	[195], [191], [96]

2.4 Representation Learning Approach

In the representation learning approach, features are learned implicitly rather than explicitly from profile, content, and network. The implicit learning of features is made possible by implementing methods for learning network embeddings. These network embeddings are inherently low dimension representation of network nodes.

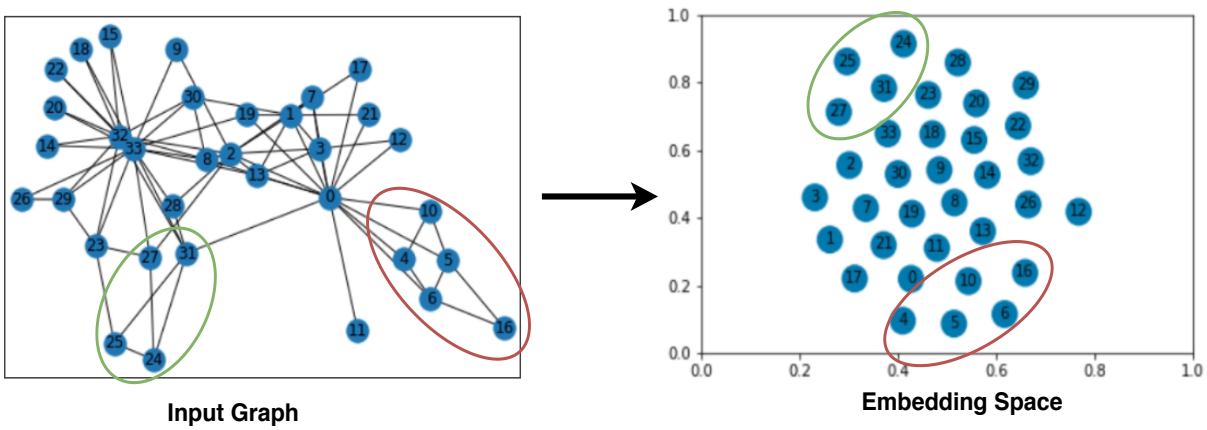


Figure 2.4: Illustration of representation learning in which low dimension node embeddings are learned. The input graph is modeled based on the Karate Club comprising of 34 users (nodes) and their interactions (edges). Representation learning converts these nodes into low dimensional embedding vectors (two dimensional space in this illustration) such that neighbors in input graph are closer to each other in the embedding space.

As an example, we depict Figure 2.4, the standard Karate club modeled as a network graph. The network science community uses this network very often for illustration, where 34 nodes represent the club's users and 78 edges represent the interactions among the users outside the club. The goal of representation learning is to transform this network graph into low dimension node embedding representations such that nodes that have structurally near to each other are also near to each other in embedding space. As evident from Figure 2.4, node IDs 27, 31, 25, and 24 are neighbors in the network graph, and therefore, their corresponding node vectors are closer to each other in embedding space. The same can be concluded for other groups of neighbors, for instance, node IDs 4, 5, 6, 10, and 16. These low dimension representations are the features learned, which is an alternative to the approach where hand-crafted features are computed explicitly. Recently, there are a few works that have emerged which address the User Identity Linkage (UIL) problem using the approach of network embedding. We categorize these works into two main categories, namely, problem-independent and problem-dependent approaches.

2.4.1 Generic Embedding Approaches

These approaches aim to learn a generic node representation without optimizing for any specific problem. In other words, their goal is construct representations without optimizing them for the specific UIL problem. Rather, the objective is to learn low dimension effective node representations in low dimensions, using mostly the structural information present in a graph. The reason we study these works is that many of the approaches (as we shall discuss in the next sub-section) that focus on identity linkage problem draw inspirations from the optimization frameworks proposed in

these works. Given that these works do not directly focus on the user identity linkage problem, we discuss only a few well-known works in this category. Depending upon the kind of information used for learning node embeddings, we divide the works in this section in two parts, one which uses only structural information and second, which uses both structure and content (semantic) information present with nodes in the network.

Network Based

Tang et al. [148] proposed a framework, referred to as LINE for network embedding in large graphs. Their approach can be applied to different types of graphs, namely undirected, directed, and weighted. They preserved first order node proximity, which means nodes that are directly connected with each other have their embeddings closer than other nodes. Besides, they also preserved second-order node proximity, to capture the notion that related nodes can also be present at two-hop distance. In order to make stochastic gradient descent based optimization computationally feasible, they proposed negative edge sampling technique to learn the embeddings at a faster rate, thereby ensuring the LINE works well on large scale graphs. Perozzi et al. [124] proposed the DeepWalk framework to learn node representations in a given network. The key difference from LINE was the adoption of an alternative approach to learn node embeddings. They performed random walks over the graph in a truncated manner and leveraged the concept of skip-gram models typically used in language modeling to learn latent representations of nodes in a graph. The nodes which appeared in the truncated random walk were considered to be closer (or similar) to the starting node from where the walk started. Wang et al. [158] proposed SDNE (Structural Deep Network Embedding) method, which departs from the earlier methods based primarily on shallow methods. Given that network structures were complex and non-linear, SDNE learns node embedding using a semi-supervised deep learning approach. As a result, non-linear relationships in graph structures were captured in the SDNE approach. In order to take care of sparsity and preserve network structure, the SDNE framework leveraged first and second-order node proximity as proposed by prior works like LINE. Grover et al. [47] extended the notion of a random walk proposed in the DeepWalk framework [124] by introducing biased-ness in the random walks. They proposed node2vec framework for learning node features in a given network. The notion of biased-ness captured the diversity in the network neighborhood. More specifically, the biased walk controlled the graph exploration strategies, whether to walk in a depth-first manner or a breadth-first manner. They introduced a new parameter, referred to as search bias which is used to control the exploration of a random walk. Chen et al. [21] proposed PME (Projected Metric Embedding) model. As per this model, they learnt the node embeddings and their relationship embeddings in separate embedding space. They projected node embeddings onto the relations embedding space and then measured the relationship proximities. For optimization, an adaptive sampling approach that is loss-aware was employed. Matsuno et al. [106] solved the user identity linkage by recasting a network into

multiple layers. More specifically, they modeled social networks as multiplex networks representing multiple layers, each of which depicted a specific type of relationship. They proposed the MELL (Multiplex network Embedding via Learning Layers) framework, which is an embedding method for multiplex networks. MELL converted each node in each layer into low dimensional vectors and then leveraged edge probabilities to learn node embeddings in the multi-layer scenario.

Network & Content Based

Methods discussed till now leverage only the structural information in a network to learn node embeddings. However, there are works which, in addition to the network information, also utilize the semantic relationships between nodes to create node representations. Xu et al. [168] proposed two embeddings, one based on structural proximity of nodes, and another based on the semantic similarity. More specifically, they considered two types of links, namely structural-close links and content-close links, to capture structural closeness and common interests. Liang et al. [89] proposed Dynamic User and Word Embedding model (DUWE) that observed the relationship between words used in user generated text over time. Both user and word embeddings were learned in the same embedding space, thereby effectively capturing their similarities. The learned embeddings help in the retrieval of top-k most relevant users with given interests. Like Xu et al. [168], this work also captured both network and content proximities in the given network. Liu et al. [92] presented a Self-Translation Network Embedding (STNE) framework that was based on sequence-to-sequence translation models taking into consideration both network and content features of the node. They performed random walks to generate sequences. The goal of the STNE framework was to translate content sequence to node sequence.

2.4.2 Problem Specific Approaches

In this section, we discuss prior works that learn embeddings of low dimensions focusing on a specific problem, which in our case is to find user identities across social networks belonging to the same person. Like the categorizations in the previous section, we divide prior works in this section as well based on the type of information used to learn node representations.

Network Based

Liu et al. [94] proposed an Input-Output Node Embedding (IONE) framework whose goal was to perform alignment of user identities by learning node representations which preserve the relationship of follower-followee. IONE framework brought the embedding vectors of nodes closer in embedding space who have similar followers and followees. To capture follower-followee relationship, they defined input and output context for each node. Input context defined the contribution of a given

node to each of the neighbors of the node. Output context defined the contribution of neighbors of a given node to the node. For learning node representations, they used negative sampling with stochastic gradient descent. Man et al. [102] introduced a framework referred to as PALE (Predicting Anchor Links via Embedding), which predicted anchor links via embeddings. They use few known linked identities referred to as anchor links as supervisory information. First, it created a low-dimensional representation for a given social network. Then, they followed it up by building a matching function that is trained using the known anchor links. Sun et al. [147] addressed the issue of lack of labeled data and the unavailability of seed anchor node pairs. They proposed a bootstrapping approach that labels node pairs that are likely to belong to the same user in an iterative manner. A network of users is represented as a knowledge graph, and the process of assigning labels is referred to as entity alignment. Chu et al. [25] proposed CrossMNA, which referred to the cross network embedding method. They address the issue of linking users across multiple social networks rather than two social networks only. CrossMNA used only the structural information of nodes to create node embeddings. They used two types of information, namely intra-vector, which reflects structural information inside a given network and inter-vector, which captured the common-ness among the potential node pairs belonging to the same user. Yasar et al. [172] proposed a Global Structure Assisted Network Aligner (GSA-NA) method. Rather than using local information, they leveraged global structure present in graphs to align nodes belonging to the same user. From the given set of anchor nodes, they identified a small subset of anchors referred to as vantage points, which act as reference points for large graphs. Instead of working on the entire graph, computations were performed on these vantage points, thereby reducing the computational costs considerably. Yang et al. [171] proposed Graph-Aware Embedding Method (GAEM), which modeled the relationships between two or more social networks into a single unifying framework. They used only the network's structure information to learn node embedding for the user identity linkage problem. For second-order structural similarities, they made use of the K-nearest neighbor algorithm to identify nodes at second order proximities. Cheng et al. [24] proposed USAIP (User Alignment via Structural Interaction and Propagation) model which captured the information interactions among users in a structural manner. USAIP can learn from the new structural information formed by newly added nodes in the network along with existing structural information. Zhou et al. [192] proposed an approach based on deep learning, referred to as *DeepLink*, which learnt node representations based on network structures.

Network and Attribute Based

Heimann et al. [56] proposed the REGAL framework, which performs graph alignment based on representation learning and used cross-network matrix factorization method (xNetMF) for optimization. To speed up the computations, they employed approximations of dense and large matrices, which are of low-rank, as proposed by Drineas et al. [33]. Each node was represented as

a vector that was formed from structural information and attribute information available in the node. A combined node similarity function that captured attribute-based distance and structure-based distance is employed. Su et al. [146] proposed MASTER framework to overcome the three shortcomings of robustness, comprehensiveness, and multiplicity in the prior works. The MASTER framework worked across multiple social networks and combines information from node structure and node attribute information. They propose constrained dual embedding (CDE) model that simultaneously align more than two social networks and learn node embeddings at the same time. Zhang et al. [188] aimed to address the problem of diversities in the node neighborhood and error propagation by proposing MEgo2Vec node embeddings. It was based on graph-based neural networks to represent the immediate neighborhood of nodes across two social networks. Attribute information associated with each node was considered as a list of words. They converted each word into embedding vector and subsequently create character embeddings using CNN. A combined objective function that concatenates the difference between structure embeddings and attribute embeddings was employed.

Network and Content Based

Wang et al. [164] proposed LHNE mode referred to as Linked Heterogeneous Network Embedding model. It created a unified framework to leverage content and structure posted by users for node representation learning. From the content posted by users, they extracted the topics representing user interests using Latent Dirichlet Allocation (LDA). For the structure, friend based node proximities were preserved across the social networks. They learnt a joint optimization function combining interests and friends' information. Sajadmanest et al. [133] proposed CRMP (Connector and Recursive Meta-Path) framework, which is a meta-path based approach. In addition to the actual friendship network, they created a content based network taking into account location, keywords, and time of the post. They projected friends information and post information on a heterogeneous graph and meta-paths captured walk on user nodes and content nodes in such a graph. Nechaev et al. [115] proposed a graph embedding framework to link users in the knowledge base (DBpedia) with Twitter users. They constructed co-occurrence matrices using the words present in content posted by users. For constructing graphs, they considered retweet and mention behavior on Twitter. Xie et al. [167] proposed an unsupervised approach to perform user identity linkage based on the concept of factoid embedding. A factoid is a triple containing two users and the relationship between them. For instance, a user following another user. Their approach learnt factoid embedding by taking into consideration that each user has diverse attributes, content updates, and neighborhood. Zhou et al. [193] presented *TransLink*, similar to the approach followed by Sajadmanest et al. [133]. They created a network based on text, location, and time of user generated content.

Network, Content, and Profile

Very recent works like the INFUNE (INformation FUsion and Neighborhood Enhancement) framework for fusion of information and enhancement of neighborhood proposed by Chen et al. [22] leveraged network, content, and profile information belonging to the user. They used all the heterogeneous information describing users to generate user node embeddings through encoder-decoder models. For evaluation, they employed hit-precision as the metric to find linked user identity in top-k candidates.

2.5 Research Gaps and Future Directions

In this section, we discuss some research gaps and directions for future work in the context of users joining multiple social networks. Prior works address most of the problems in social networks in the context of a single social network by monitoring user behavior in one single social network. However, with the availability of linked user identities, more comprehensive information about user's behaviors over several social networks can be obtained [101]. Therefore, some recent works have begun exploiting this comprehensive user information obtained by linking user identities, however, there is scope for more work to be done.

2.5.1 Recommendations

Making recommendations by using user's behavioral preferences on more than one social network is an important application. Ozsoy et al. [119] collected data from different online platforms, namely Twitter, BlogCatalog, Facebook, Flickr, LastFm, and YouTube to help in recommendations. They compared recommendation systems built from only one social network with those built using many social networks and found that recommendations done using user data from multiple social networks were more robust and comprehensive. Ostuni et al. [118] and Musto et al. [114] performed recommendations by leveraging Linked Open Data (LOD) platforms like DBpedia. However, most of these prior work makes use of data-level linkages across the social network. It would be interesting to explore in the direction of user identity linkage to improve user recommendations.

2.5.2 Link Prediction

In the context of two or more social networks, the problem of link prediction helps in finding out whether a user would join a new social network or not. Zhang et al. [183] presented a survey of prior works that focus on link prediction across social networks. More specifically, they focused on user-user links and user-location links across social networks as well for the prediction tasks.

Zhang et al. [187] also proposed meta-path based approach for collective link prediction across multiple social networks. Qi et al. [128] proposed to solve link prediction in the presence of sparse connectivity of users in a given network. In such a scenario, they made use of the inter-connections in other social networks of the users to help in link prediction. While there are prior works which predicted links across social networks, we need to extend the idea beyond links. More specifically, predicting the social behavior of users by leveraging their behaviors in multiple social networks.

2.5.3 Social Capital of User

In the context of online social networks, the social capital [70, 145] of users refer to their popularity and acceptance in the social network world which prior works have measures in different ways in terms of likes, shares, engagements, and followers that users receive. Quantifying social capital is helpful for many applications like influence prediction and propagation in the political domain [82] and human resource management [61]. Zafarani et al. [176] studied variations in popularity and friendship for the same users across different social networks. They used this information to predict if a given user is going to be popular on a target social network or not. Besides this work, most of the other prior works have quantified social capital by using only a single social network. There is a need to measure a user's social capital using that engagement received by the user across multiple social networks.

2.5.4 Social network forensics

Malicious users perform online crimes, and very often they leave behind digital footprints across social networks [101]. Michel et al. [109] proposed an ontology based methodology for the detection of salient traits of users across social networks, which can help in cyber forensics. In a typical scenario, a user who indulges in online crime on a particular social network would not leave any identification trace in the network where the crime was committed. However, if we can link that user's identity to another social network where his behaviors are more apparent, then it would help in tracing the culprits. Given the widespread prevalence of cybercrimes, more work in this direction needs to be done.

2.5.5 User Privacy

There are privacy implications on users owing to the linkage of their identities across social networks. As we know, some OSNs provide access to the professional network (like LinkedIn) while others provide access to a more personal network (like Facebook). Managing one's identity on multiple such OSN platforms are tricky. A user would typically post about her personal life related events on a social network like Facebook, but would probably not do so on a professional network like LinkedIn.

In other words, a user tries to maintain different contexts on different OSN platforms. With online social networks, there is a collapse of user context [31,156], which has privacy implications. Elias et al. [35] performed a detailed study on the implications of OSNs on the personal and professional life of users, particularly learners in educational settings. On the other hand, using a personal network in the professional domain comes with its share of challenges. Ranieri et al. [129] studied the use of Facebook by teachers for professional purposes. Fox et al. [39] investigated the challenges faced by professionals, particularly teachers, in managing their personal and professional identities in social media. Besides, there are other factors as well that complicate and affect users' participation in these networks. For instance, a friend request received on a professional network would be accepted even if a requester is not personally known whereas, on a network where user shares her personal events, such a friend request would likely be turned down. However, when a user's identity is linked across such social networks, then it gives rise to a variety of privacy implications which are seldom addressed or acknowledged. It would be worthwhile to explore the impact of user identity linkage on users who are conscious about their privacy.

2.5.6 Dataset Biases

A number of data collection approaches, which we discuss in Section 2.2, have been used in the past to collect user identities belonging to the same user across social networks. Each of those approaches relies on specific characteristic behaviors of users who maintain identities across multiple social networks. Consequently, behavioral biases exhibited by users often get manifested in these linked identity datasets. Dataset biases, in general, are being extensively studied. For instance, in the domain of computer vision, there are several prior works [58, 150, 151] that investigate the biases in image datasets. However, the study of behavioral biases that manifest in the linked user identity datasets has not been explored. Such a study will ensure that the learned models are free from biases and are more robust to different kinds of the dataset being used for their training.

Having discussed the background, related work, and future directions in this chapter, we shall discuss our contributions in details in each of the next chapters in this thesis.

Chapter 3

Data Collection Methods

Having discussed the background and related work, in this[1] chapter we discuss several data collection methods employed in the context of our problem of User Identity Linkage (UIL). To recall, users create their accounts on multiple Online Social Networks (OSNs) to access a variety of content and connect to their friends. Consequently, user behaviors get distributed across many OSN platforms, and the goal of the UIL problem is to link user identities belonging to the same person. We refer to the user identities belonging to the same person as *linked user identities*. To collect user information in a comprehensive manner, an important step is to collect user accounts (identities) of the same individual across multiple OSNs.

3.1 Background

One of the key motivations behind the collection of linked user identities is *user profiling*. We refer the systematic approach to performing a large scale collection of user behaviors across OSNs as *user profiling* [37]. We recall a few advantages and applications of user profiling. Users tend to provide incomplete information on a single social network, either with purpose or otherwise. Knowing the same user's identity on other social networks would help in the comprehensive profiling of the user in terms of user's profile, user's content, user's behavior, user's preferences, and friends. In the advertising world, it enables targetted advertisement [170] and improved recommendations. Prior works have studied most of the problems in social networks like information propagation, link prediction, algorithmic biases, discrimination studies, and community detection in the realm of a single social network, which we can now investigate across multiple social networks. In social media crimes and cybersecurity problems like cyberbullying, fake accounts, and spamming, we are often

[1]Work presented in this chapter is mostly taken from our published paper. **Rishabh Kaushal**, Vasundhara Ghose, and Ponnurangam Kumaraguru. Methods for User Profiling across Social Networks. In *Proceedings of the 12th IEEE International Conference on Social Computing (SocialCom), 2019.*

looking for user footprints within the same social network in which incident occurred. If the user's identities on other social networks are known, it is only going to help in the investigation [62]. From the user's privacy standpoint, individuals can be shown their comprehensive profiles and likelihood of linkage of their identities and nudged to control their online behavior so that their digital footprint decrease [72]. Lastly, there is no agreed benchmark dataset in the problem domain of identity resolution. So, large scale data collection of linked identities would help researchers compare and evaluate their proposed solutions.

Given the importance of social profiling, a lot of emphasis has been given in research community to solve the problem of User Identity Linkage (UIL). Data driven approach to solve the problem has two key steps. Firstly, a large number of user identity pairs belonging to *linked identities* and *non-linked identities* are collected. Secondly, machine learning based model is constructed over the user behavioral features extracted from user identities. In this chapter, we focus on presenting and implementing methods for the collection of linked identities across OSNs. As depicted in Figure 3.1, initially there are two social networks namely $SN1$ and $SN2$. Some user identities belong to the same person, they are referred to as *linked identities* while other identities belong to different individuals; we refer them as *linked identities*. After we apply a data collection method, depicted as $M_1, M_2, ..., M_n$, some of the linked identities get detected, while some remain undetected. In the *first part* of this chapter, we focus on five data collection methods to obtain linked identities namely Operator based search or Advanced Search Operator (ASO), Social Aggregator (SA), Cross-Platform Sharing (CPS), Self-Disclosure (SD) and Friend Finding Feature (FFF). Taking all these methods together, we collect[2] linked identities of 238,042 individuals across 43 different OSNs. Subsequently, in the *second part* of this chapter, we present a detailed quantitative and qualitative assessment of these methods. For *quantitative assessment*, we evaluate the number of

[2]Refer at http://precog.iiitd.edu.in/resources.html for dataset details.

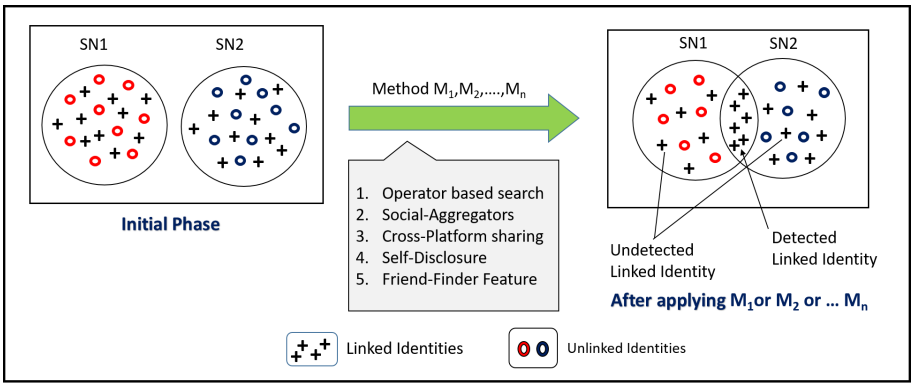


Figure 3.1: Visual depiction of progressive stages in which linked identities are collected starting from no linked identities and gradually progressing to collect as many of them as possible by applying methods for user identity linkage.

social networks covered by a method and number of linked identities obtained per-individual across OSNs. For *qualitative assessment*, we leverage standard parameters, namely data completeness, consistency, accuracy, validity, availability and timeliness, inspired from ISO 9000:2015[3] and the works of Scannapieco et al. [134] and Loshin et al. [98].

To the best of our knowledge, we are the first to focus exclusively upon the methods for collecting linked identities, which is the de-facto first step for user identity linkage. Key contributions of our work are:

- Detailed description and implementation of data collection methods to retrieve linked identities, thereby facilitating user identity linkage.
- Comprehensive evaluation of data collection methods, both qualitatively and quantitatively.
- Step towards creation of a comprehensive dataset that we can use as a benchmark dataset for user identity linkage research.

3.2 Data Collection: Methodologies & Implementations

Collecting user data from online social networks has always been a challenge, and given that the data is related to users, there are privacy issues as well [105]. Application Programmer's Interfaces (APIs) offered by OSNs have restricted their capabilities [57] over the past years due to data privacy concerns. The data breach [12] involving Facebook and Cambridge Analytica has added more challenges in terms of data collection even for academics to collect data for research purposes [48, 83]. With all this happening, users are becoming even more privacy-aware [15, 137], which would dissuade them from mentioning all details in their accounts, resulting in *missing values* when data is collected. To make things worse, social network platforms like Twitter allow users to change their account handles. There are several reasons which makes users change their usernames [66], ranging from keeping their identities hidden to no specific reason. The username changing behavior further complicates the data collection process.

A generic framework for user profiling, as depicted in Figure 3.2, comprises three steps, namely data collection, data integration, and data extraction & indexing. The first step is **data collection** in which we identify the source of data followed by a selection of data collection methods. In the second step, we follow by **data integration** in which we store user identities collected from all methods at a single data store point, which we refer to as *Linked Identity Data Store (LIDS)*. Finally, in the third step, we perform **data extraction & indexing**, which involves collecting the three components of user identity, namely profile, content, & network. Next, we describe each data

[3]International Standards Organization: https://www.iso.org/standard/45481.html

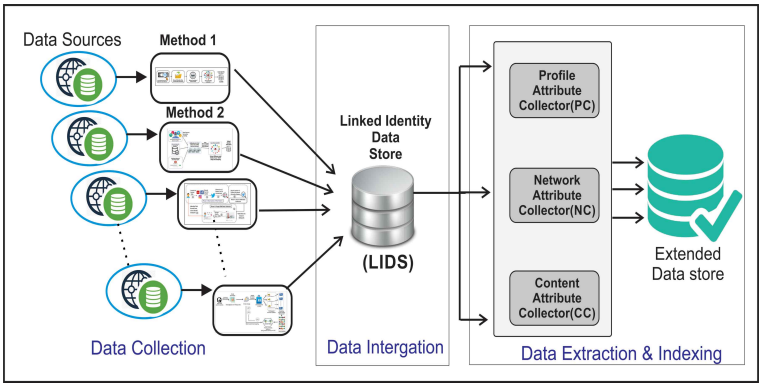


Figure 3.2: Generic Framework for User Profiling. We collect linked user identities from OSN platforms using several methods and store in LIDS. Subsequently, we obtain profile, network, and content information from user identities depending on API support of OSN platform provider.

collection method in detail. Next, we present a detailed methodology adopted to perform data collection using five methods, which is the focus of this chapter.

3.2.1 Advanced Search Operator (ASO)

Search engines have played a pivotal role in the information age [34]. Irrespective of the search engine provider, the basic steps involved are [103] (1) Crawler: which collects information in terms of web pages from across the web. (2) Storage: which stores (indexes) the collected information such that it's retrieval are quick, and (3) Query: which provides an interface to the user where they can enter their queries for retrieval of desired information. For user querying, search engines provide *advanced search operators* [54] using which we can obtain more detailed and specific information. In this work, we leverage Google's advanced operator search [13, 132], to obtain information of specific type. To give an example of advanced operator search, if a user types the following search query `intext:facebook.com,twitter.com filetype:xlsx`, then Google search engine would locate all web documents that have `facebook.com or twitter.com` written as text anywhere in the document with the additional constraint that these documents must be of `xlsx` file type. In the information security community, analysts and hackers use these advanced search operators to extract personal and private information about users (like user emails, passwords, and credit card details) stored in different file types on web servers, this technique is referred as *google dorking* [149] or *google hacking* [104, 181].

In the context of our problem, we leverage the google-dorking approach to retrieve files that contain linked user identities. Specifically, we are interested in `csv` or `xlsx` files whose each row contains social media handles belonging to the same person. We emphasize here that we did not explicitly link the user identities in this method. The third-party entities link the user identities, and the linked user identities recorded in the files present on the websites indexed by search engines. As

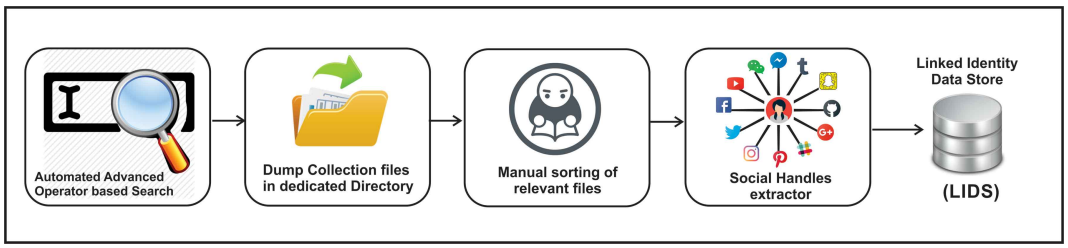


Figure 3.3: Pipeline for Advanced Search Operator (ASO) method. We perform search using advanced operators to retrieve files containing linked user identities.

outlined in Figure 3.3, we implement the following steps to retrieve such files.

1. We run a script using Selenium tool which contains pre-configured search queries on Google. [4] We write search queries such that two or more social media handles are present as *text* in the file and the format of the file is `csv` or `xlsx`.

2. We parse the search results, and extract the desired files.

3. We read the files manually to identify files which indeed contain social media handles on multiple OSN platforms belonging to the same user.

4. Finally, we extract the social media handles corresponding to the same individual and save them in LIDS.

We may note that this method relies upon mis-configuration of web servers where users' information is kept in files and folders crawled and indexed by web search engines. To the best of our knowledge, we are the first to explore google dorking for obtaining linked user identities.

3.2.2 Social Aggregator (SA)

There are several websites on which users register and themselves provide details of their social media handles on well-known social networking websites such as Twitter, Tumblr, Facebook, Flickr, LinkedIn, Pinterest, and YouTube. We refer to such websites as *social aggregators*. These websites enable users to perform self-presentation [59] and increase the visibility of their presence across multiple social networks. Many prior works [42, 93, 122, 180, 185] leverage this user behavior of social aggregation to obtain linked user identities across social networks. One such website that we use to collect linked user identities is *about.me*[5] and use it to collect linked user identities. Users put one-page descriptions introducing themselves by giving details of their social media profiles along with a background image and abbreviated biography.

[4]Selenium https://www.selenium.dev/, a tool that automates user browser activities.
[5]About.me: https://about.me/ is a site that offers registered users to link their multiple online identities.

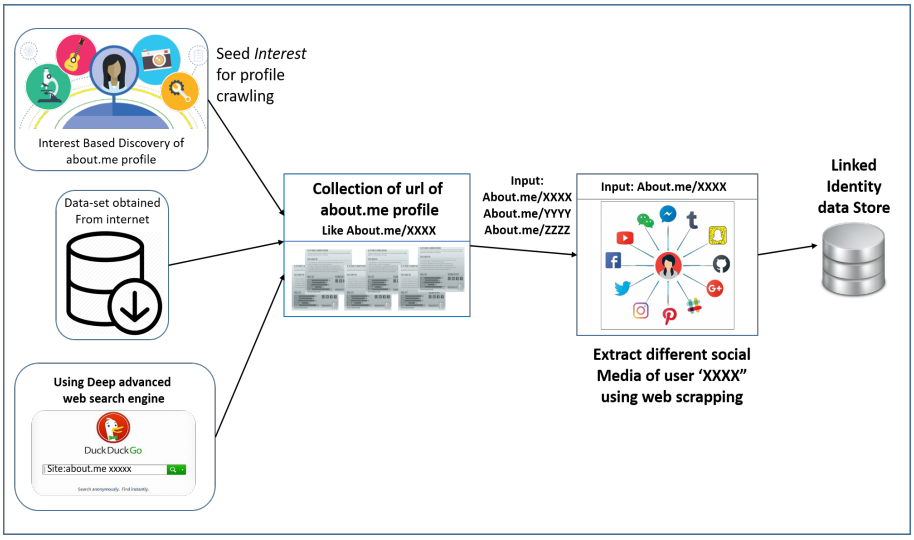


Figure 3.4: Pipeline for Social Aggregator (SA) Method. We obtain linked identities using three approaches namely, discovery feature, an external dataset, and Google dorking.

As depicted in Figure 3.4, we implement three different approaches to collect linked user identities from about.me website.

1. Discovery Feature: Initially, when we started data collection, *about.me* provided an option to search user profile using topic-based search (referred as *discovery feature*). Given an interest topic as input, it would return all the user profiles having that interest.

2. External Dataset: After one month of data collection, in March 2018, this discovery feature of *about.me* got discontinued. On exploring further, we added a public dataset[6] containing *about.me* profiles which we used for our further work.

3. Google Dorking: Lastly, we leverage the advanced search operator method based on google dorking. We used interests as *intext* and *site* as *about.me* to obtain more user profiles.

It may be noted that above steps are applicable for social aggregator *about.me*, for other platforms, the steps and challenges would differ.

3.2.3 Cross-Platform Sharing (CPS)

Several OSN platforms provide users with an option to share their content across other (target) OSN platforms. Given that users have their presence on multiple OSN platforms, they are motivated to share their content with all their friends in these different OSN platforms. We refer to this user behavior as *Cross-Platform Sharing* (CPS) behavior. Jain et al. [67,68] and Correa et al. [28] have

[6]http://scholarbank.nus.edu.sg/bitstream/10635/137403/2/about_me.sql

used this approach of cross-posting, referred as *self-mention*, to collect identities belonging to the same person.

In Figure 3.5, we implement a case study of Instagram-Twitter social network pair, and explain the scenario of user's cross-posting behavior and our approach for data collection in the following steps:

1. A user makes a post on a social network (referred to as *source*, in this case, Instagram) from her mobile phone.

2. After making the post, the user selects the sharing option on the post made, and shares the post on another social network (referred to as *target*).

3. Given that sharing is done using the source OSN's mobile app, the shared content on target OSN appears with a specific URL pattern $\backslash instagram.com\backslash \mathtt{p}\backslash$ in the *text* of the post which points to the post made on the source network.

4. For the data collection, we rely on the API provided by the target network (in this case, Twitter), to search for posts that contain such pattern. In addition, we also check the `source` field in the *json* object of the post. This check guards us against the scenario where a user cross-posts someone else's post.

5. We parse the collected post, extract the URL pattern specified above, and expand the URL to reach to the original post.

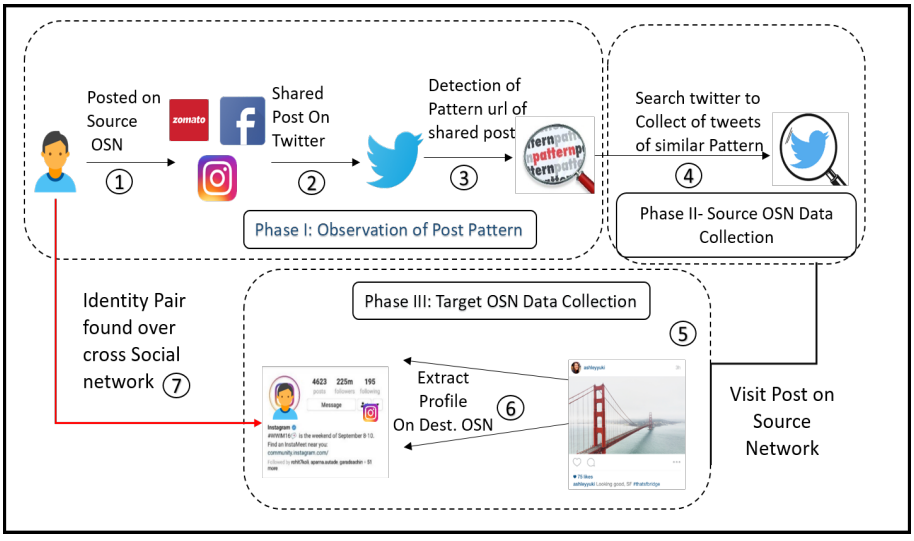


Figure 3.5: Pipeline for Cross Platform Sharing (CPS) method. We depict a case study performed on Instagram-Twitter social network pair. User makes a post on Instagram, and then shares it on Twitter.

6. From the original post, we either use API or scrap the posting page to obtain the user's identity on the source network (Instagram).

7. Finally, in the last step, we link the user's source network identity (Instagram) with the user's target network identity (Twitter).

It may be noted that the above steps makes few assumptions, (1) target source network has support for API to perform search in posts, (2) cross-platform shared post has a unique URL pattern on target social network, (3) there is a `source` field that points to the source social network.

3.2.4 Self-Disclosure (SD)

Whenever a user signs up on OSN, there is an option to provide a user description in the profile settings page. In the user description, there is a provision to specify details of their identities on other OSN platforms. We refer to the user behavior who provide details of their social media identities on other OSN platforms as *self-disclosure*. Many prior works [23,88,140,175,189] leverage this user behavior to collect linked user identities.

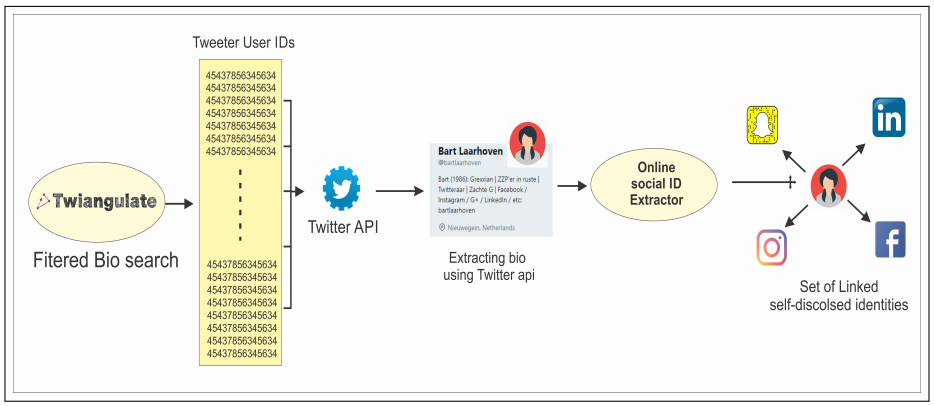


Figure 3.6: Pipeline for Self-Disclosure (SD) Method. We use Twiangulate to perform bio-field based search for Twitter users. Then, extract URLs in bio field to other social network identities maintained by the user.

In our work, we perform our implementation on the social network, Twitter. On the user's profile page on Twitter, there is a *bio* field where the user provides a short free-form text describing themselves. As depicted in Figure 3.6, we perform the following steps:

1. Twitter API doesn't provide an option to search in the `bio` fields of the users. So, we use *Twiangulate* web tool[7] to perform `bio` search and collect all those twitter profiles which has at least one social network mentioned in their `bio` field.

[7]Twiangulate: http://twiangulate.com/search/

2. Then, we create a regular expression to detect diverse patterns in the bio field on Twitter because a user can mention details of their social media handles in a variety of ways. For instance, a user can mention *TV Host and Media Trainer - Instagram: @NeshanTVxyz Snapchat: @Neshaxyz* while another user can use acronyms like *TV Host and Media Trainer - IG: @NeshanTVxyz SP: @Neshaxyz FB: nashbin123*. To handle these variations, we tokenize all text and check for URLs occurrence, which could lead to other OSN platforms.

We note that the above steps would work only for those users who *self-disclosure* their identities on other OSN platforms.

3.2.5 Friend Finder Feature (FFF)

Whenever a user joins a new OSN, we sign up using our unique identifier, say email or phone number. This information is used by OSN to find our friends in our email contacts or phone contacts. The OSN platforms use this information to offer *friend finder* option to help connect to those friends who already have an account in OSN. Goga et al. [41] leveraged this mechanism of *friend-finder* in social networking sites to collect linked user identities.

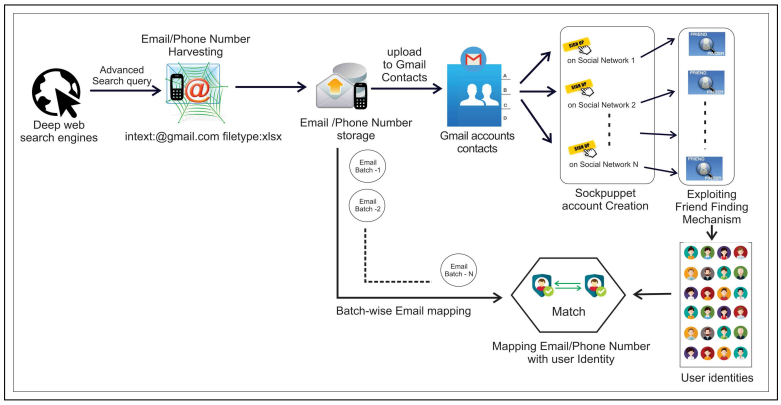


Figure 3.7: Pipeline for Friend-Finder Feature method. We use searches to harvest emails, then create a Gmail account and add collected emails in contact list. Subsequently, we use this Gmail account to join multiple OSN platforms to use their friend-finder feature.

As specified in Figure 3.7, we implement the following steps:

1. Email Harvesting in Deep Web: Polakis et al. [126] proposed harvesting of emails using social networks. However, due to increasing restrictions in APIs of the social network, we explored other methods, one of which was to use advanced search operators as a query in the deep web search engine like Duckduckgo to retrieve emails.[8]

[8]Deep Web: www.duckduckgo.com

2. Email Account Creation & Sockpuppeting: Next, we create an email account and add the harvested emails in this email's contact list. We do this to create the impression that these emails in the contact list belong to people known to the email holder.Sockpuppets [100] are user accounts created for online deception. They are used to facilitate the promotion of business, generate favorable reviews, and so on. In our case, we use the newly created email to sign-up on several OSN platforms.

3. Friend Finder: After we sign-up on different OSN platforms, we leverage the *friend-finder* mechanism on those platforms which suggest friends based on emails in the contact list of our email used to sign-up, besides other methods.

4. List Matching: Once we obtain a list of suggested friends on different OSN platforms, we use user name based string matching techniques to find identities belonging to the same user.

We note that that this method assumes that users typically use the same email to sign-up at different OSN platforms to enable email-based friend suggestions.

3.3 Dataset Description

In this section, we summarize the data collection based on our implementation of the five methods described previously. Table 3.1 provides details on the number of linked identities collected using each of the data collection methods.

Table 3.1: Results of data collection methods implemented in this work. Data collection in each of them is continuing and numbers are increasing by the day.

Data Collection Method	**Linked Identities Collected**
Cross-Platform Sharing (CPS)	104,233
Self Disclosure (SD)	69,815
Social Aggregator (SA)	53,692
Advanced Search Operator (ASO)	9,802
Friend Finder Feature (FFF)	500
Total Linked Identities	238,042

Among all the five methods, Cross-Platform Sharing (CPS) method yielded the maximum number of linked identities (104,233), keeping Twitter as the target network and source networks being Facebook, Instagram, and Zomato. Social Aggregator (SA) method using *about.me* gave 53,692 linked identities taking into account all three approaches followed in it, namely discovery feature, which contributed 15,973, a dataset which added 15,620 and search engine based, which yielded 22,099. Self Disclosure (SD) method, which extracted identities by parsing *bio* field of Twitter profile using Triangulate, gave 69,815 linked identities. We collected 9,802 identities using Advanced

Search Operator (ASO) queries on *google*. Lastly, using the Friend-Finder Feature (FFF), we could obtain 500 linked identities due to a lack of harvested emails. In the next sections, we compare our implementations of five methods, both quantitatively and qualitatively.

3.4 Quantitative Evaluation

For quantitative evaluation, we evaluate data collection methods based on two metrics, namely social network coverage and per-user linked identity count. We use the data described in Table 3.1 and present the results after applying the pre-processing step of de-duplication.

3.4.1 Social Network Coverage

One of the goals in a data collection method is to collect linked identities across as many OSN platforms as possible. For our work, we define *social network coverage* as the number of OSN platforms on which the given data collection method was able to collect linked identities. By design, some data collection methods like Cross-Platform Sharing (CPS), are limited to giving only a pair of social network identities, so we ignore such methods for this metric.

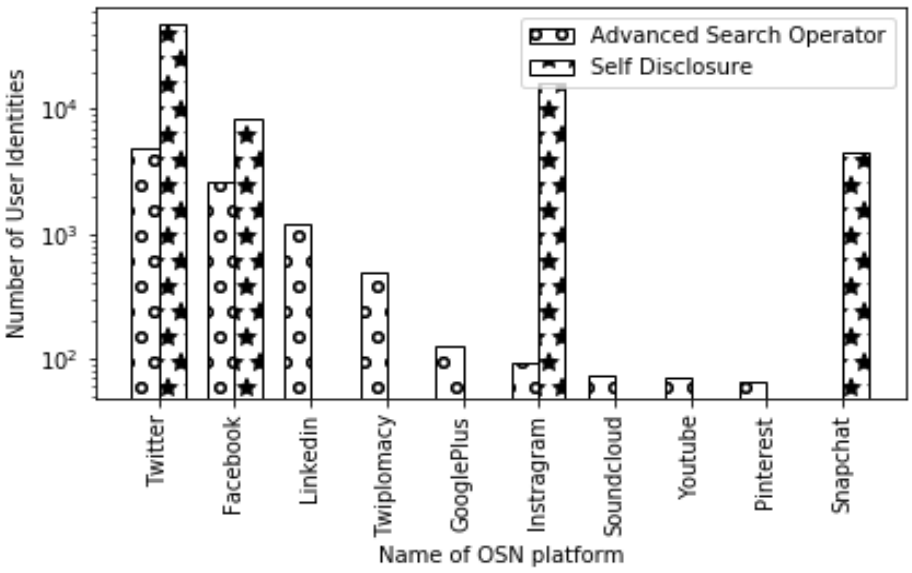


Figure 3.8: Distribution of coverage of OSNs on which linked identities were collected using Advanced Search Operator (ASO) and Self Disclosure (SD) methods. Values on Y-axis are raised to the power of 10.

As depicted in Figure 3.8 Advanced Search Operator (ASO) method covers nine social networks Facebook, Twitter, Youtube, Linkedin, Google+, Pinterest, Instagram, Soundcloud, and Twiplomacy. To recall, the ASO method exploits the vulnerability in web server, whereby, files containing

linked user identities are exposed to web crawlers. Therefore, the presence of user identities across nine social networks indicates potential leakage because this information is not obtained owing to explicit user consent but rather a vulnerability in web server hosting their social media information. The coverage of the Self Disclosure (SD) method includes four social networks Twitter, Facebook, Instagram, and Snapchat. This indicates that users tend to self-disclosure only a few of their social media handles on `bio` field of their Twitter accounts.

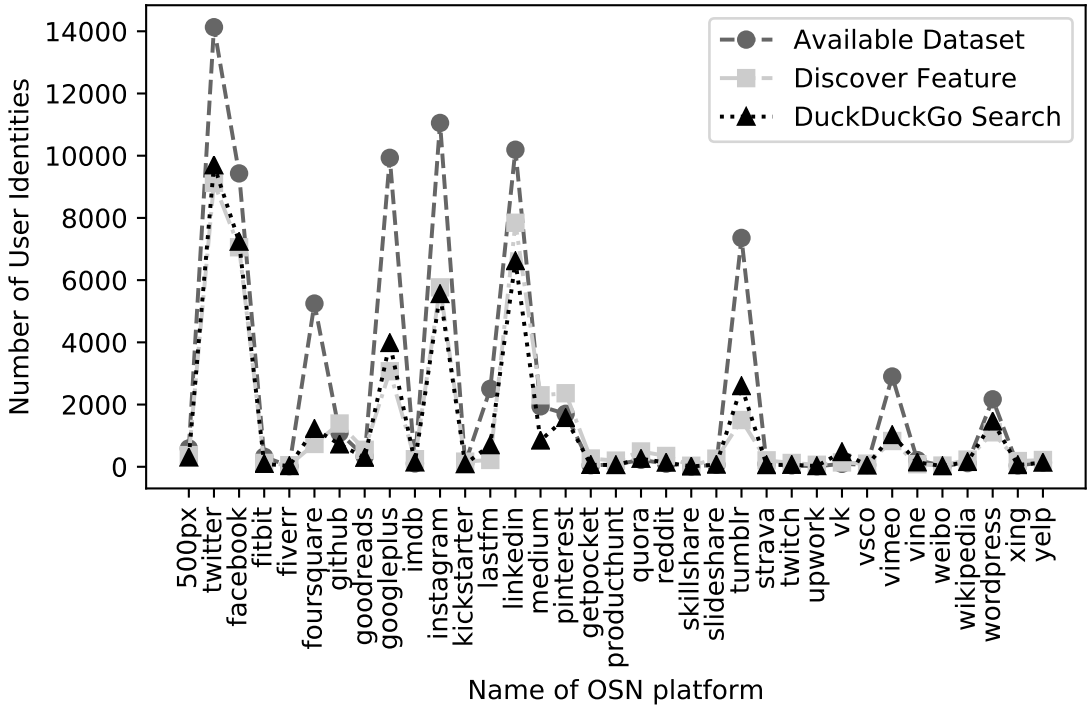


Figure 3.9: Distribution of social network covered using Social Aggregator (SA) method for collection of linked identities. This method by far is the best in terms of OSN coverage.

Further, in terms of OSNs coverage, Social Aggregator (SA) method performs the best. While we obtain user identities across 43 OSNs, however, in Figure 3.9 we plot only those OSN platforms where number of user identities obtained are greater than 50. This suggests that users link their identities across multiple OSN platforms when they create their profiles on social aggregators to self-promote their presence on multiple sites. Among the three approaches employed in the SA method, the one that leverages search engine (*duckduckgo*) gave the best results. This indicates that a combination of two methods, namely, advanced search operator and social aggregation, can yield better results.

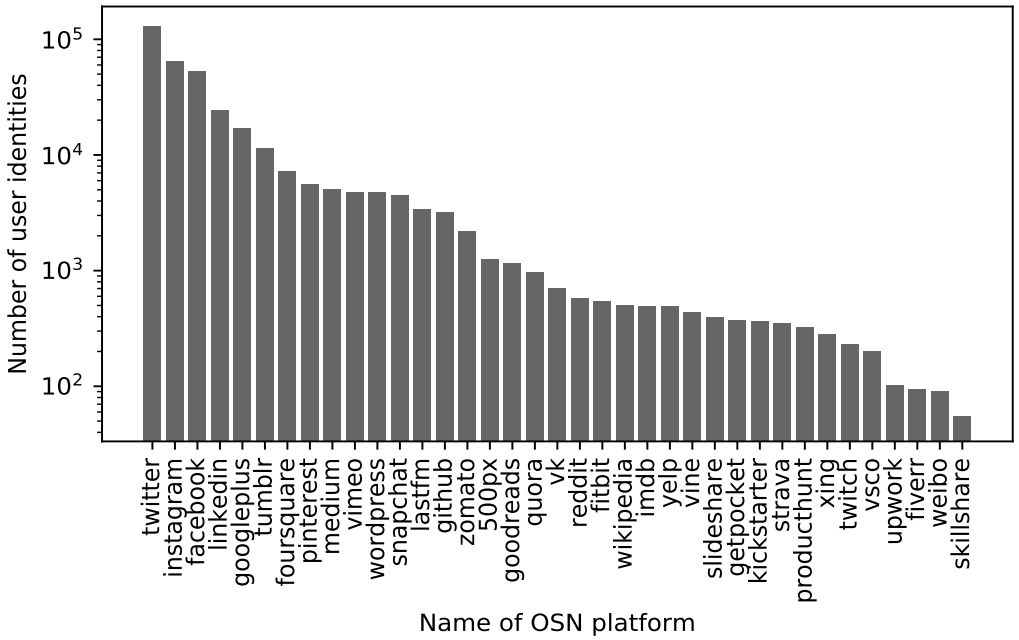


Figure 3.10: Distribution of social network covered using all methods for collection of linked identities. Although we obtain data for 43 OSNs, we plot only those OSN platforms for which we got at least 50 user identities or more.

Taking into account all the data collection methods, we obtain linked user identities across 43 OSN platforms. In Figure 3.10 we depict the number of linked identities across those OSN platforms for which we obtained more than 50 user identities. As evident, we obtain more than 10,000 user identities on Twitter, Instagram, Facebook, LinkedIn, Google+, and Tumblr. And for the OSN platforms namely FourSquare, Pinterest, Medium, Vimeo, WordPress, Snapchat, Last.fm, GitHub, Zomato, GoodReads, and 500px we get more than 1,000 user identities. However, there are many OSN platforms, for which we obtain less than 500 user identities, so the issue of data sparseness does exists. Nevertheless, to the best of our knowledge, this is by far the best social network coverage in the literature of user identity linkage.

3.4.2 Per-user linked identity count

We define *as per-user linked identity count* as the number of linked identities found for each person in the context of our problem of user identity linkage. In other words, the number of OSN platforms on which we have been able to obtain a given user's identities. While it is true that obtaining higher per-user linked identity count is dependent on user behaviors, it is also true that some of their user behaviors (like cross-platform sharing) have an inherent constraint that they would return at most only two social identities at a given time.

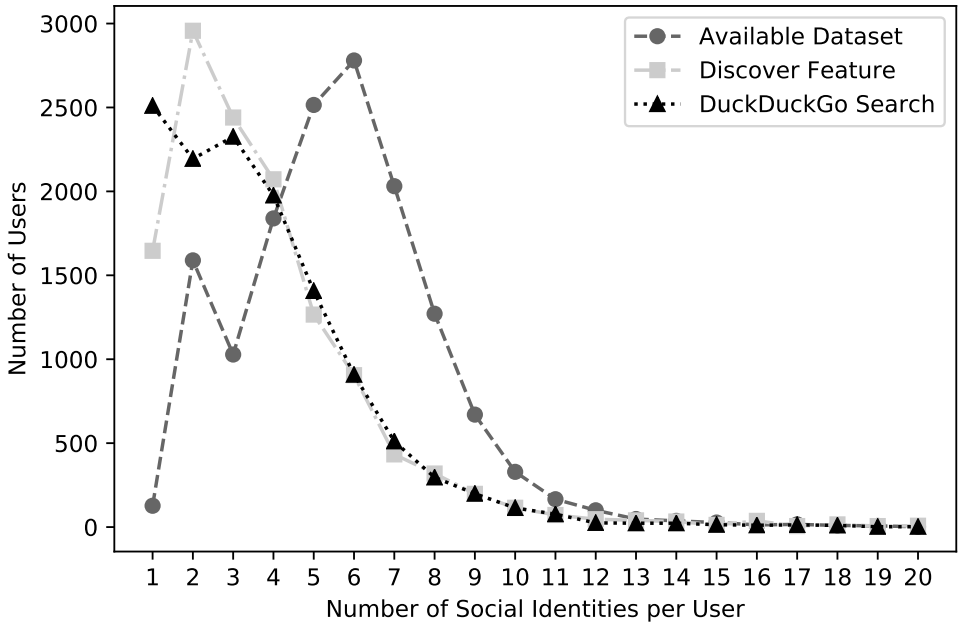


Figure 3.11: Distribution of per-user linked identity count using Social Aggregator (SA) method for collection of linked identities.

Figure 3.11 shows the per-user identity count distribution for Social Aggregator (SA) method. The discover feature, and public dataset approaches gave better results, while the search engine's approach gave comparable results with discovery features when per-user identity count increased beyond 5.

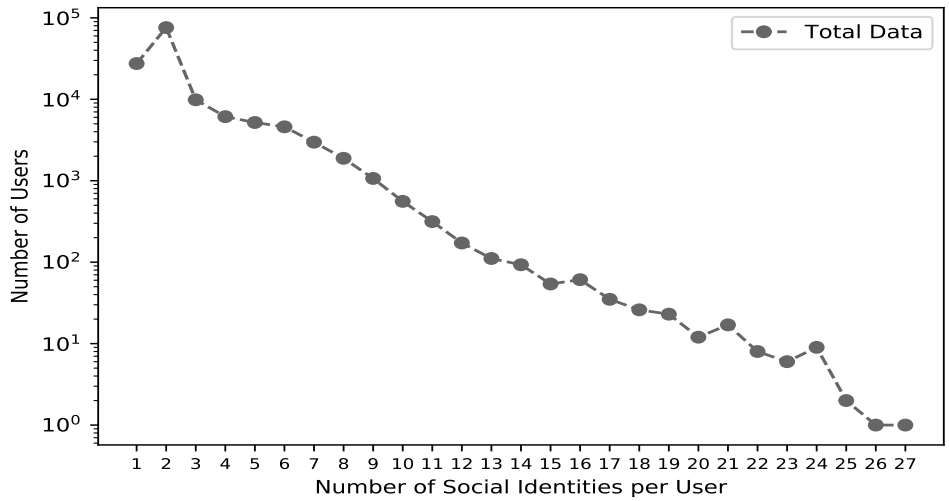


Figure 3.12: Distribution of per-user identity count across multiple socials obtained taking together all methods for collection of linked identities.

In Figure 3.12, we depict the per-user identity count obtained by taking into account all the data collection methods. It is evident that as the number of identities per user is increasing, the number of such users is decreasing. Furthermore, this establishes that users maintain two or more identities across multiple OSN platforms.

3.4.3 Identity Pairs, Triples, and Quadruples

In this section, we describe our data collection mentioned in Table 3.1 and present our final results of data collection performed through all the methods discussed earlier. Please note this data is presented before the pre-processing step of de-duplication.

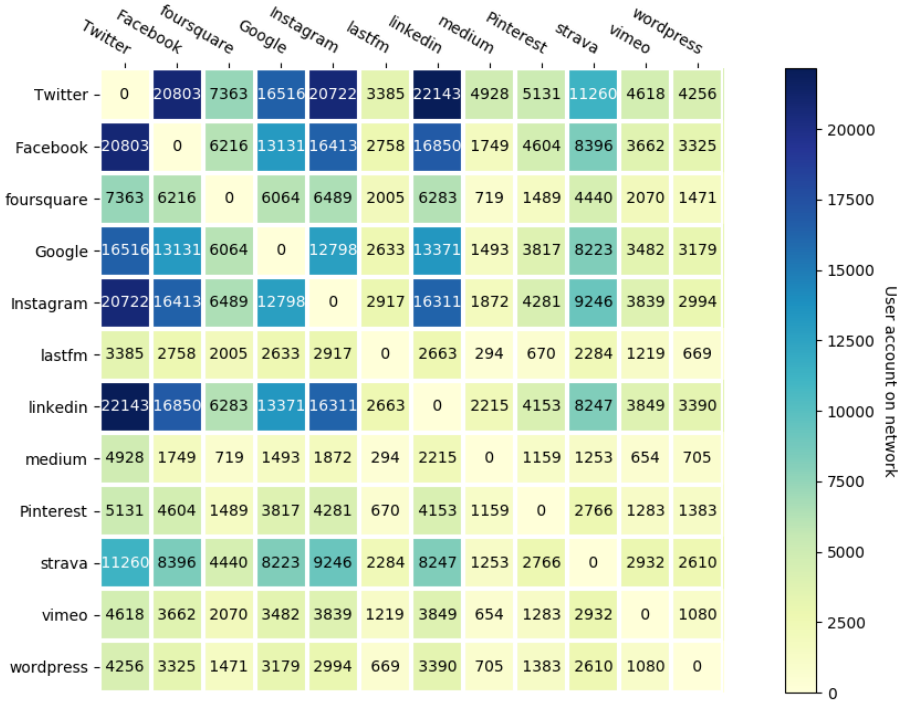


Figure 3.13: Two Dimensional matrix depicting linked identities between a pair of two social network, only those OSN platforms are considered where we have obtained more than 5,000 user identities.

We present the total pairs of linked identities collected in Figure 3.13 for social networks, namely Twitter, Facebook, Instagram, Linkedin, Strava, Lastfm, Vimeo, Wordpress, Google, Pinterest, Medium and Foursquare. Twitter-LinkedIn and Twitter-Facebook with 22,143 and 20,803 user identity pairs are among the largest social networks on which linked user identities are found. This is followed by LinkedIn-Facebook, Instagram-Facebook, and LinkedIn-Instagram comprising of 16,850 pairs, 16,413 pairs, and 16,311 pairs, respectively.

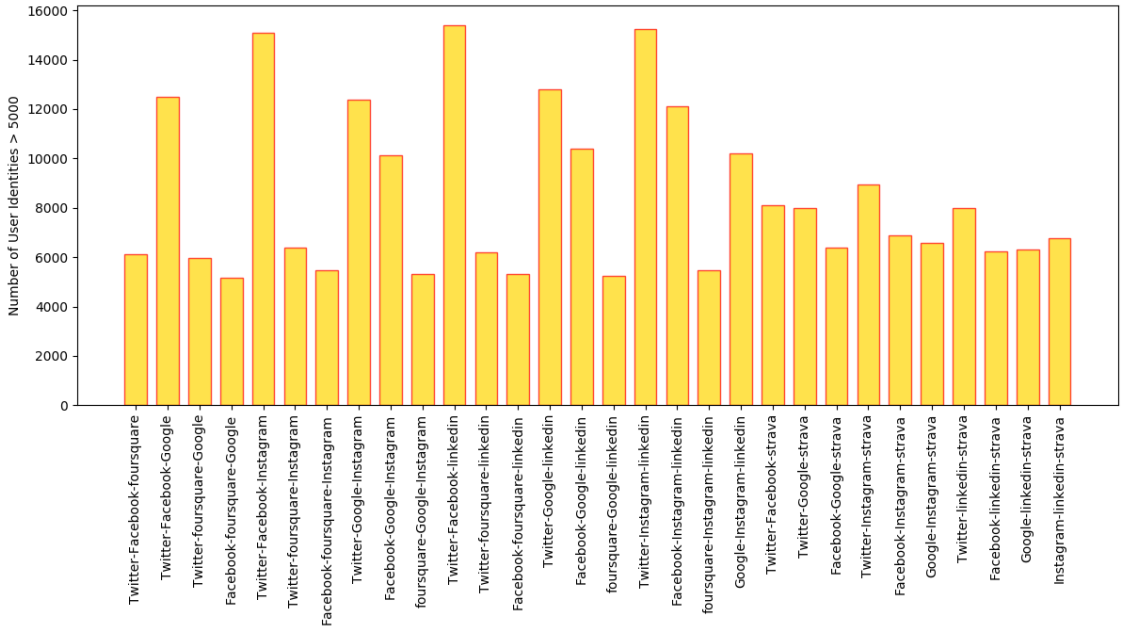


Figure 3.14: Distribution depicting all triples found between three social network, only those OSN platforms are considered where we have obtained more than 5000 user identities.

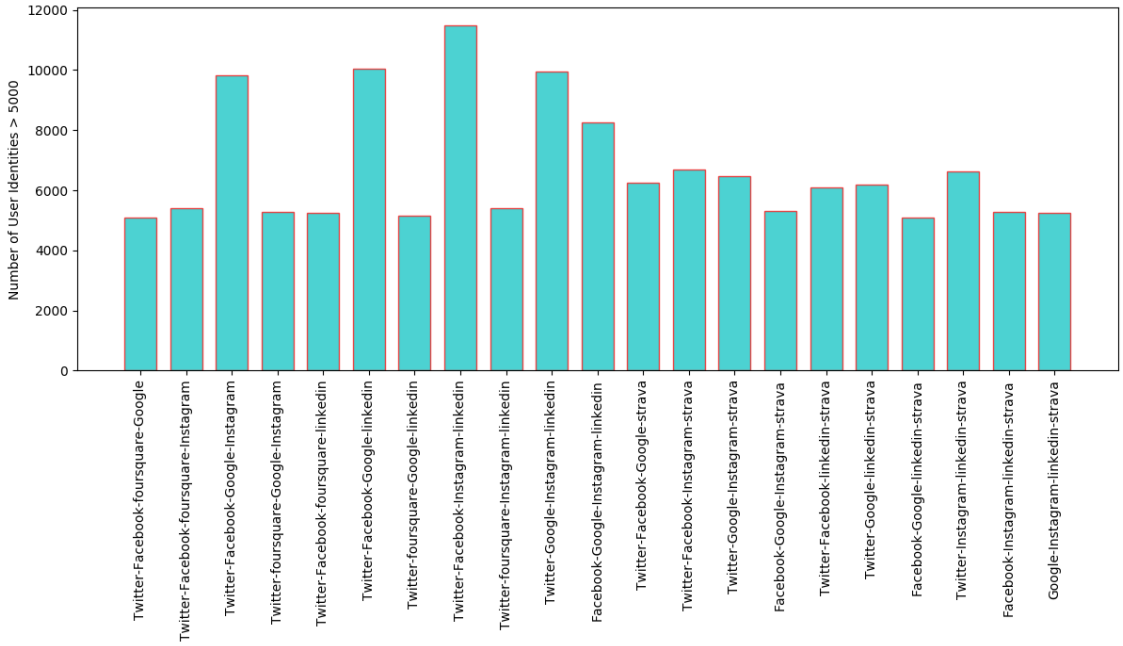


Figure 3.15: Distribution depicting all quadruples found between four social network, only those OSN platforms are considered where we have obtained more than 5000 user identities.

In Figure 3.14 and Figure 3.15, we present triples and quadruples of social networks where we obtained linked user identities. Among the triples, the ones giving a large number of user identities are Twitter-Facebook-Instagram, Twitter-Facebook-LinkedIn, and Twitter-Instagram-LinkedIn. Among the quadruples, Twitter-Facebook-Instagram-LinkedIn gives the maximum number of linked user identities.

3.5 Qualitative Evaluation

Inspired by the works of Scannapieco et al. [134] and Loshin et al. [98], we leverage metrics from ISO 9000:2015[9] standard for quality assessment, namely, completeness, validity, consistency, accuracy, and timeliness to qualitatively evaluate the effectiveness of different data collection methods to obtain linked data identities. It may be noted that each method has inherent assumptions and dependencies. For instance, in the data collection methods, namely SD, SA, and CPS, we rely on user-generated content. We assume that users have generated the content genuinely without any falsifications.

3.5.1 Completeness

We define completeness in the context of our problem, as the ratio of collected linked identities of a user by a given data collection method to the actual linked identities across all OSNs for the same user. From an information retrieval perspective, this is similar to *recall*, and it is a characteristic of a given data collection method. Ideally, the methods should obtain all linked identities, but in practice, it is not possible, refer Table 3.2 in which we explain our qualitative observations on data collection methods from the point of view of completeness metric.

3.5.2 Validity

Validity in the context of linked identity collection would mean whether the steps involved in the data collection method would continue to work (in other words, would remain valid) in each run at subsequent times. Validity is an essential criterion for *reproducibility*, refer Table 3.3 for our explanations in terms of validity of various data collection methods based on our qualitative assessment. It may be noted that users who are privacy conscious would not want to self-link their identities. So, if and when they find that their behaviors could link their identities, so they can undo those behaviors to prevent linkage in future.

[9]International Standards Organization: https://www.iso.org/standard/45481.html

Table 3.2: Qualitative Analysis of Data Collection Methods w.r.t. completeness metric.

Method	Completeness	Remarks on Completeness
ASO	Medium	It depends on the number of social identities submitted by the user to a given server whose data is crawled by search engines.
SA	High	It depends on the number of social identities self-promoted by the user on social aggregation websites.
CPS	Low	By design, in this method, we get only a pair of user identities across two social networks provided the user indulges in cross-platform posting.
SD	Medium	It depends on the number of URLs directing to the user's social identities on other OSNs mentioned by the user in his/her account description.
FFF	Low	It depends on the availability of the friend-finder feature on the OSN and friends have registered with the same email address on that OSN platform.

Table 3.3: Qualitative Assessment of Data Collection Methods based on Validity.

Method	Validity	Remarks on Validity
ASO	High	It depends on the support for advanced search operators by search engines, and as long as web servers allow search engines to crawl files and folders stored in them.
SA	Medium	It depends heavily on the API support or feature (like discovery in *about.me*) supported by the social aggregator.
CPS	Medium	It depends on the cross-platform sharing support by source network, presence of URL pattern, and ability to search it on the target network.
SD	Medium	It depends on API support provided by the social network to retrieve profile attribute (like `bio` field in Twitter).
FFF	Medium	It depends on the friend-find support provided by the social network.

3.5.3 Consistency

In the context of the problem of user identity linkage, we define consistency as the ability of the data collection method to return the same results each time we execute it. These data collection methods depend on the search and retrieval mechanisms employed by the data provider; therefore, these internal mechanisms would affect the results in each run. In Table 3.4, we outline our observations on the data collection methods based on consistency metric.

Table 3.4: Consistency based Qualitative Analysis of Data Collection Methods.

Method	**Consistency**	**Remarks on Consistency**
ASO	Low	Results heavily depend upon the mechanisms of crawling, indexing, and ranking of search results by the search engine.
SA	High	As long as the user doesn't change her social media handle details on social aggregators, the results would remain consistent.
CPS	High	Expect similar results (linked identity pairs) as long as the user continues to do cross-platform sharing.
SD	Medium	It depends on indexing and search results ranking of profile attribute based (say `bio` in Twitter) search on the social network.
FFF	Medium	It depends on the friend recommendation algorithms employed by social networks.

Consistency of the collected data, in our case, the linked user identities, would also depend heavily on support for reconfigurability in user profiles provided by social networks. For instance, users can change their *username* on Twitter and Instagram. Consequently, if a user changes her *username*, then the previously obtained username-pair for this user would become stale and inconsistent.

3.5.4 Accuracy

In the context of our user identity linkage problem, the accuracy of the data collection method would mean their ability to provide correct results, in other words, the linked user identities obtained indeed belong to the same person. All the methods rely upon information made available by users; therefore, as long as user-provided information about their social media identities is correct, the data collection method would work well. Social Aggregator (SA) and Self Disclosure (SD) methods are directly dependent on the information provided by the user. As long as user-supplied information is accurate, the data collection methods are guaranteed to return true positive linked identities, for details refer Table 3.5 for our observations.

Table 3.5: Our explanations for Degree of Accuracy for Data Collection Methods.

Method	Accuracy	Remarks on Accuracy
ASO	High	It depends on the correctness of the data entered into web servers by users at some point in time, which got crawled and indexed by search engines.
SA	High	Depends on the correctness of social media identities entered by users on social aggregator websites.
CPS	High	Works well as long as a specific URL search pattern is present and `source` field points to the source social network.
SD	High	It depends on the correctness of social media identities entered by users on profile attributes (say `bio` field) in social network websites.
FFF	Medium	Most often, OSN platforms suggest only the names and profile pictures of friends through the friend-finder feature of social networks; hence there is a possibility of wrongly associating social identities to the same person.

3.5.5 Timeliness

For our problem, timeliness would mean whether a data collection method can find linked identities for a given user on demand, otherwise return false. Table 3.6 presents details of our qualitative assessment on the response of the data collection method to the timeliness metric.

Table 3.6: Qualitative Analysis of Data Collection Methods based on Timeliness Metric.

Method	Timeliness	Remarks on Timeliness
ASO	Low	Given a user identity, it is very less likely that we can get other identities belonging to the same person using ASO unless we perform a large scale data collection.
SA	Low	The same as above also holds for social aggregators, because not many users would have created their profile on such social aggregator sites.
CPS	Low	Assuming that the social network platform provides API support to find targets of cross-platform shared posts.
SD	Medium	It depends on whether the user has mentioned his social media handles in their profile (say `bio` field in Twitter) of a social network.
FFF	Medium	If we know the email address of the given user identity, then finding other social media identities across other social networks is possible using the friend-finder feature.

3.6 Discussions and Future Work

In this chapter, we presented a common framework for the collection of linked user identities and implemented five data collection methods. We compared and evaluated them qualitatively and quantitatively. Based on our experience from the implementation of these methods, we list down a few suggestions for prospective researchers who would want to work in the domain of user identity linkage. Social Aggregator (SA) method is useful in the scenario when we want to study user behavior across a large number of OSNs, in other words, for better OSN coverage. Self Disclosure (SD) method would yield decent coverage of OSNs but in a limited manner. On the contrary, if one has to target only a specific pair of OSN, then Cross Platform Sharing (CPS) method would be the best option. Advanced Search Operator (ASO) method would be useful if we want to target popular social networks (like Facebook, LinkedIn, Twitter, etc). Friend Finder Feature (FFF) is practical only when one has a large pool of emails or phone numbers of users, for instance, a service provider; otherwise, the scale would be small. FFF would also be useful in the scenario when one has to investigate an unexplored social network.

There are a few limitations to our work. In all the data collection methods, we obtained the linked identities identified by *only* the *usernames* belonging to the same person across multiple OSN platforms. Obtaining information like posts made by the user and friends of the user would be more useful but is quite challenging because it is heavily dependent on API support provided by social networks, which is decreasing by the day, owing to user privacy considerations. In terms of future work, say for Social Aggregator (SA) method, we have investigated *about.me*, it would be interesting to extend it over other similar social aggregation platforms like Google+. Similarly, in Advanced Search Operator (ASO) method, we may go beyond *google* search engine and explore other search engines like bing, duckduckgo, etc. In Cross Platform Sharing (CSP) method, we have taken Twitter as the target social network, and we can extend to include other OSNs as well. Similarly, we parsed only Twitter's *bio* field in Self Disclosure (SD) method. We may explore other social network platforms.

For ethical reasons, all data collection methods in this chapter rely on the availability of public data and, in most cases depends on user behaviors where users themselves have made their details *linkable* either using social aggregation website or self-mentioning their social media identities on their profile pages and so on. However, users may not be aware of the implications of public availability of their data. For users who are privacy concerned and would not want their identities to be linked, we recommend that (1) they should not cross-post content across OSNs, (2) they should not provide details of other OSNs on their social media profile pages, (3) they ought not to use the same email to register at different OSN platforms and not provide their social media details on websites that allow them to crawl by search engines. For the users who are at the other end of the spectrum and who want to increase the visibility of their actions, we recommend that

they create their profiles on social aggregators, cross-post often, explicitly link their social media handles, and register using the same email address on different OSN platforms. However, regardless of the kind of user, this work helps towards building a system that can help users understand the amount of their own data that is available and can be collected so that they remain more careful and safe online.

Chapter 4

User Identity Linkage DataSet Biases

In this[1] chapter, we focus on the biases that exist in the data collected using the different data collection methods discussed earlier. To recall, in the scenario of multiple social networks, the problem of User Identity Linkage (UIL) is to find whether a pair of user identities on two social networks belong to the same individual or not. Prior works (as we explain in Chapter 2) collect linked user identities (identities on different social networks belonging to the same person) using several data collection methods as we describe in Chapter 3. In this chapter, we refer to the linked identity data, thus collected as *user identity linkage dataset*. To collect this dataset, we leverage user behaviors in different social networks, so behavioral biases get manifested in the dataset. In this chapter, we perform a detailed investigation into these dataset biases, a work that has mostly remained under-explored in the identity linkage research community. More specifically, we find that behavioral biases in the datasets manifest in the form of lexical differences in user-generated content, particularly in usernames and display names configured by users. We characterize, detect, and quantify these biases on more than one million user identity pairs obtained by leveraging two user behaviors, namely cross-posting and self-disclosure. We observe that users who self-disclose their usernames and display names on different social networks show higher lexical similarity than users who cross-post. These behavioral biases lower down the performance (precision and recall) of learning models by 5-20%. Inspired by discrimination measurement metrics, we propose and implement a framework to quantify the extent of these biases and find that 15-20% of test data get affected.

[1]Work presented in this chapter, is mostly taken from our published paper. **Rishabh Kaushal**, Shubham Gupta, and Ponnurangam Kumaraguru. Investigation of Biases in Identity Linkage DataSets. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing (SAC), 2020.*

4.1 Introduction

Recall that in the *user identity linkage* problem, our goal is to find whether two input user identities belong to the same individual or not. If the two user identities belong to the same individual, we refer them as *linked identity pair* else *non-linked identity pair*. Linked identities present more comprehensive coverage of user behavior, thereby, helping in better recommendations. Prior works, as we discuss in Chapter 2, address the UIL problem in two steps, as shown in Figure 4.1.

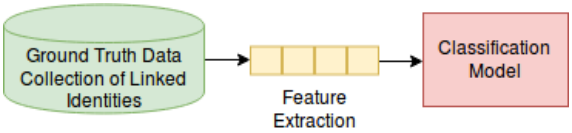


Figure 4.1: Basic Framework for User Identity Linkage which comprises collection of ground truth linked identities, feature extraction, and construction of classification model.

The first step involves the collection of linked identity pairs on two OSMs using a well-defined data collection method, which we refer to in this work as a *data source*. The second step involves learning of a data-driven classification model over handcrafted features extracted from the three dimensions of a user identity namely profile information [88,175], content posted (and interacted) [41] and the friend network [194]. To recall, this is the most common formulation of the UIL problem, as we described in Chapter 2.

More formally, given two identities I_a and I_b from two social networks a and b, respectively, the goal is to learn a *classifier function* f as defined in equation 4.1, such that it returns 1 if I_a and I_b belongs to the same individual else it returns 0.

$$f(I_a, I_b) = \begin{cases} 1, & \text{if } I_a \ \& \ I_b \text{ belong to same user.} \\ 0, & \text{otherwise.} \end{cases} \tag{4.1}$$

Numerous data collection approaches, which we discuss in Chapter 3, have been proposed in the past to collect linked user identities. Each of them relies on the typical behaviors of users who maintain identities across multiple social networks. As a consequence, the behavioral biases exhibited by users get manifested in these identity linkage datasets. Given that we learn classifier function on features derived from identity linkage datasets, the biases inherent in the datasets affect these models as well. Although biases have been extensively studied particularly in the image datasets [150, 151], however, the study of behavioral biases that manifest in the linked user identity datasets have not been explored. In this chapter, we fill this gap by investigating the *impact* of behavioral biases in linked user identities on the performance of an identity linkage solution by addressing three research questions.

1. **Detection:** Does user behavioral bias exist in identity linkage datasets collected using dif-

ferent data collection methods?

2. **Implication:** Whether the performance of an identity linkage model is affected by behavioral biases in the dataset?

3. **Quantification:** Can we measure the extent to which these behavioral biases are manifested in the identity linkage datasets?

To address these questions, we consider two different approaches for collecting linked user identity pairs (in other words, two *data sources*), based on two user behaviors, namely cross-platform sharing (CPS) and self-disclosure (SD), which we describe in Chapter 3. In CPS user behavior, we find users to occasionally share their post made on one social network (source) across other social networks (target), which we refer to as *cross-posting*, as depicted in Figure 4.2. This is typically done to provide wider visibility to their content across their networks on different social networks. However, in the process of this cross-posting, the users eventually also end up linking their identities on both source and target social networks, as described in Section 3.2.3 in Chapter 3. In this work, we consider Instagram as a source social network, and Twitter as a target social network, the dataset thus obtained is referred to as *CPS dataset*. In self-disclosure user behavior, we look for users who

Figure 4.2: CPS User Behavior: User makes an Instagram post, then shares the same post on Twitter. Link to the Instagram post appears on Twitter post (tweet).

explicitly mention (or *self-disclose*) their identities on other social in their profiles on the source network, as we explain in Section 3.2.4 in Chapter 3. In this work, we consider Twitter as the source social network on which users self-disclose their identities on other social networks by configuring the `bio` field of Twitter, as shown in Figure 4.3. We look for only the Instagram social network in the `bio` field of Twitter (so that it is comparable with CPS dataset in respect to the social network pair); the dataset thus obtained is referred to as the *SD dataset*. This is typically done by users to keep their followers on Twitter informed about their other social identities so that those interested may connect to them on other social networks as well.

Besides these user behaviors which we leverage for linked identity collection, we also focus on two elementary behaviors that a user performs for maintenance of their identities on a social network, namely *username creation* and *display name configuration*. We choose these user behaviors for two

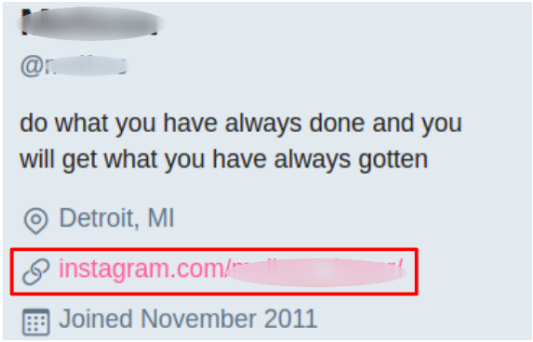


Figure 4.3: Self-Disclosure: Instagram identity is mentioned in the bio-field of Twitter identity.

main reasons. First, on both Twitter and Instagram, it turns out that users have the freedom to choose the username and display name to be used for their identities. Second, in terms of attribute availability, usernames and display names are always inherently present in all user identities, and they are also publicly visible. We employ numerous similarity metrics (referred to as *features* as we shall discuss later in Section 4.5.2 of this chapter) to measure the lexical similarity of username and display names configured by users on Twitter and Instagram. We find that linked user identities on Twitter and Instagram obtained by leveraging SD behavior exhibit far more significant lexical similarities than those obtained by CPS behavior. Given that users are free to configure their usernames and display names on Twitter and Instagram, we infer these lexical differences as the presence of *user behavioral biases* in user identity datasets.

Further, to see the impact of these biases on the performance of the identity linkage model, we follow the typical approach adopted by prior works [41,88,175,194] in building a learning function based on lexical features derived from usernames and display names. Ideally, we would expect a machine learning model (in our case identity linkage model) to be *generalizable* [157] in the sense that model performance does not get affected by the data source. However, our work shows that the model trained on CPS dataset and evaluated on SD dataset (and vice-versa) performs 5-20% poorly in terms of precision and recall than the model which is trained and evaluated on the same dataset. This clearly indicates that behavioral biases that exist in the dataset have an adverse effect on the performance of the model.

For quantification of biases, we leverage the works on discrimination studies and biases [17,117,197], to propose a *novel framework* that uses discrimination discovery metrics to quantify the extent of damage caused by behavioral biases in the dataset. More specifically, we apply *situational testing* proposed by Luong et al. [99] for measuring individual-level biases to quantify the behavioral biases in user identity datasets. As per our framework, we combine the linked identities obtained using cross-posting and self-disclosure user behaviors, introduce a new attribute which we call *data_source*, whose value is set to either CPS or SD. In the context of discrimination studies, we treat this new attribute as the *protected attribute*, which enables us to apply discrimination

measurement metrics to quantify behavioral bias in user identity datasets. We find that 15-20% of test data records get affected by biases in the training dataset irrespective of the learning model employed.

- Ours is the *first* work which detects, and studies the impact of data source bias on the performance of identity linkage models.
- We propose a *novel framework* to apply discrimination studies to quantify the extent of damage caused by data source biases.

4.2 Related Literature

In this section, we discuss the prior works related to the key aspects that we leverage in our investigation of biases in identity linkage datasets. The different data collection approaches have already been discussed in previous Chapter, so we do not discuss them again here. Given that we discuss our investigations of use identity dataset biases, we provide an overview of the prior works on studying biases in datasets in general and image datasets in particular.

Biases in datasets have been extensively studied, particularly in image datasets. We discuss a few prominent works related to dataset biases in image datasets. Torralba et al. [151] highlighted the limitations of image datasets in terms of capturing the real world phenomena. They evaluated datasets on several criteria namely close world assumption, cross-dataset generalization, and relative dataset biases. Their work indicated future directions in terms of improving collection methods to avoid biases and enhancing algorithms to deal with inherent biases in the datasets. Khosla et al. [77] detected biases in datasets used for solving object recognition problem. They proposed a discriminative framework that learns two types of weights, one which is specific to a dataset, and another which are common across datasets. They concluded that it is beneficial to take into consideration biases when dealing with multiple data sources. Tommasi et al. [150] studied the difference between several datasets and performed cross-dataset analysis. They measured the performance of different debiasing approaches and discussed open issues in the field of dataset biases. Herranz et al. [58] investigated dataset biases due to the scale of objects appearing in the images. They argued that models (for instance scale-specific CNNs) that are scale-dependent give better performance in the object recognition task.

To summarize, we conclude that biases do exist in datasets owing to limitations in collection methods. And models that are explicitly made aware of these biases in datasets perform better. In the context of our problem of user identity linkage, we don't find much work in the investigation of biases. Therefore, in the next section, we propose our methodology for studying user identity dataset biases.

4.3 Proposed Methodology

In this section, we explain our approach to the following objectives.
(1) Study, detect and characterize behavioral biases in user identities on Twitter and Instagram manifested in CPS and SD datasets, and
(2) Propose a framework to quantify the severity of behavioral biases in user identity datasets.

To this end, we consider several steps as depicted in Figure 4.4. First, we apply the two data

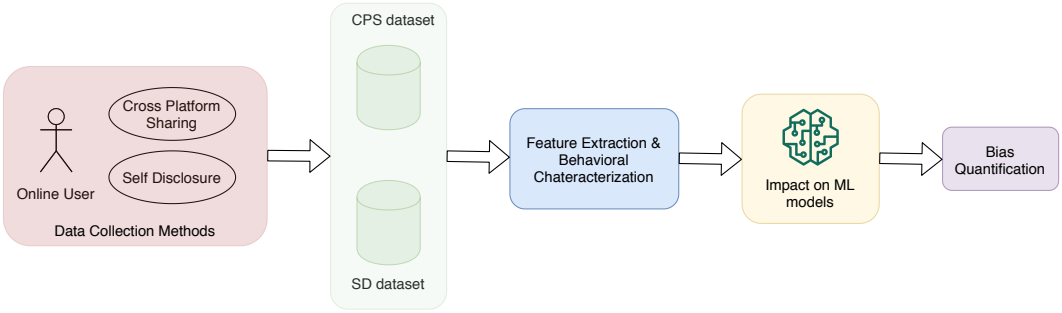


Figure 4.4: Proposed Methodology for Detection of User Identity Dataset Biases. CPS and SD datasets are collected based on user behaviors of cross-platform sharing and self-disclosure, respectively. Features are extracted and behvioral characterization done. Impact on ML models is studied, and quantification of biases performed.

collection approaches which leverage cross-posting and self-disclosure behaviors to collect linked user identities, as explained in Section 4.4. This is followed by the extraction of publicly available attributes namely user name and display name from user identities on both Instagram and Twitter. Based on these attributes, we construct features that are based upon lexical similarity metrics which are used to detect behavioral biases, as detailed in Section 4.5. Finally, we explain our approach for the quantification of user identity dataset biases in Section 4.7.

4.4 Data Collection

In this section, we explain our approaches for the collection of linked user identities (positive class, identity pairs that belong to the same person) and unlinked user identities (negative class, identity pairs that belong to different persons).

4.4.1 Linked User Identity Pairs

For the collection of linked user identity pairs, we use two data collection approaches, which are based on two user behaviors, namely cross-platform sharing (CPS) and self-disclosure (SD). Both of

these methods exploit two different behaviors exhibited by users who maintain multiple identities across social networks. Twitter and Instagram are the two social networks where linked identity pairs were collected using these methods.

1. **Cross Platform Sharing**: In this data collection method, we leverage the user behavior in which user *cross posts* i.e. share a post made on one (referred to as *source*) social network on two or more target social networks, thereby, revealing his identities on the source and target social networks. We take Instagram as the source social network and Twitter as the target social network, detailed steps are mentioned in Section 3.2.3 in Chapter 3.

2. **Self Disclosure**: In this data collection method, we look for user behavior in which users, while configuring their profile information on one social network, explicitly *mentions or self-discloses* details of identities on other social networks. More specifically, we focus on `bio` field of Twitter users to extract whether they have shared their identity on Instagram, as per details we mentioned in Section 3.2.4 in Chapter 3.

4.4.2 Unlinked User Identity Pairs

We collect unlinked user identity pairs that constitute negative samples. They are user identity pairs on Twitter-Instagram which do not belong to the same individual. We follow two approaches to generate negative samples.

1. **Random Pairing**: We generate negative samples by randomly pairing Instagram and Twitter identities obtained in the two data collection approaches. In general, if (I^1_{tw},I^1_{in}) and (I^2_{tw},I^2_{in}) are two known linked identity pairs obtained on Twitter-Instagram social networks using either of the data collection approach, then we create unlinked user identity pairs as (I^1_{tw},I^2_{in}) and (I^2_{tw},I^1_{in}).

2. **Similar Pairing**: While random pairing will guarantee us negative samples, in the real world, we do find identities that are quite similar to each, at least in terms of names. For instance, Perito et al. [122] studied the uniqueness of names and found that some names are rare, while others are quite common. To factor this, we create negative samples using this method of similar pairing. So, in this method, for a linked identity pair (I^1_{tw},I^1_{in}), we first obtain display name of I^1_{in} in Instagram and then use it to perform user search in Twitter using the Twitter Search API to find *top-k* identities on Twitter who have a *similar* display name, I^1_{tw},I^2_{tw},I^3_{tw},....,I^k_{tw}. Since (I^1_{tw},I^1_{in}) is known linked identity pair, we ignore it and keep the rest of the pairs i.e. (I^2_{tw},I^1_{in}), (I^3_{tw},I^1_{in}),....,(I^k_{tw},I^1_{in}) as unlinked user identity pairs.

4.4.3 Collected Data Summary

In Table 4.1, we give a detailed distribution of the data we collect after implementing the two methods each for generating positive samples and negative samples as discussed in the above sections.

Table 4.1: Details of linked and unlinked identity pairs obtained from different data collection methods.

Class Label	**Collection Method**	**#Pairs**
Linked	Cross-Platform Sharing	253,791
Linked	Self-Disclosure	253,791
Unlinked - Random Pairs	Cross-Platform Sharing	190,343
Unlinked - Random Pairs	Self-Disclosure	190,360
Unlinked - Similar Pairs	Cross-Platform Sharing	63,448
Unlinked - Similar Pairs	Self-Disclosure	63,454
	Total Identity Pairs	1,015,187

From each of the cross-platform sharing and self-disclosure methods, we collect 253,791 linked user identity pairs. For the negative class, we collect 190,343 and 190,360 by random pairs of identities within the CPS and SD datasets, respectively. And, we obtain 63,448 and 63,454 similar pairs using CPS and SD datasets for similar-appearing negative class identity pairs. We understand the limitation of our approach that some biases could be introduced in the negative pair sampling procedures that we adopted, however, for simplicity, we ignore these biases.

4.5 User Behavioral Analysis

Social Cognitive Theory (SCT) proposed by Bandura et al. [6] says that user behaviors are influenced by their experiences, and observing behaviors of others. In this section, we discuss three steps employed for user behavioral analysis. In the first step, we identify user behaviors that shall be leveraged for creating features. In the second step, we extract lexical features derived from user behaviors. In the third step, we perform a characterization of user behaviors based on features extracted from user behaviors.

4.5.1 Identification of User Behaviors

After collecting linked identity pairs and forming datasets, we look towards user behaviors to be used to derive features. Users exhibit a multitude of behaviors on social networks which can be categorized broadly into three types as below.

1. Content-related behaviors that entail the kind of posts are made by the user which can reveal

a user's personality traits, interests, and opinions.

2. Network-related behaviors which can be inferred from the friends maintained by the user, and people who follow the user concerned.

3. Profile-related behaviors in terms of details (like a profile picture, location, and so on) mentioned by users in their profile page, and the level of visibility provided to each of profile attribute.

Out of these three categories of user behaviors, we focus our attention to profile related behaviors. Within profile configuration, we restrict ourselves to only *username* and *display name*. In order words, we are interested to study user behavior in terms of usernames and display names that users configure in their identities across multiple social networks, particularly Instagram and Twitter. While the choice of these user behaviors appears quite restrictive, we have made this decision for three reasons. First, the support for programmatic access through APIs to content, network, and profile information of user identities in social networks has declined considerably. While Twitter does grant access, Instagram has restricted access to content and network. Second, among the various profile information, username and display name are the *elementary* profile attributes that are always configured by users and are publicly available. Third, on both the social networks Instagram and Twitter, users have the flexibility of modifying both usernames and display names, thereby making them suitable for studying user behaviors.

4.5.2 Behavioral Feature Extraction

Having decided the user behaviors to study, our next step is to define lexical similarity-based features that help us in measuring the differences and similarities in the username and display name configured by users across multiple social networks. In terms of lexical analysis, it may be observed that a username can be considered as a string and features are derived from individual characters that appear in the username. On the other hand, the display name can be considered as a set of words (strings), and features are derived at word-level. We consider the following features lexical features derived from username and display names.

- Longest Common Subsequence (LCS): For two given sequences (usernames in our case) UN^i_{tw} and UN^j_{in} from Twitter and Instagram, we find the length of longest common subsequence at the character level. For instance, length of LCS between two usernames namely $rishabhk$ and $iiit.rkaushal$ is 4 and longest common subsequence is $rsha$. Higher LCS will indicate more similarity in the username strings.

- Jaccard Distance: It is based on Jaccard similarity which considers two sets of alphabets appearing in username as input and returns their union divided by their intersection. We

compute Jaccard distance on two usernames obtained from two identities on Twitter and Instagram. For instance, jaccard distance between two related usernames *rishabhk* and *iiit.rkaushal* is 0.45 whereas for two unrelated usernames *rishabhk* and *iiit.pk* is 0.8, indicating that similar usernames have lessor jaccard distance and vice-versa.

- Normalized Levenshtein Distance: For two given usernames UN_{tw}^{i} and UN_{in}^{j}, we compute the Levenshtein distance as the minimum number of edits at the character level. Types of edits allowed are insertion, deletion, or substitution of a single character. We divide the distance by the length of the shorter username to normalize. For instance, Levenshtein distance between two usernames namely *rishabhk_* and *rk.iiit* is 8 and the shorter username length among them is 7 so the resulting normalized Levenshtein distance is 1.142.

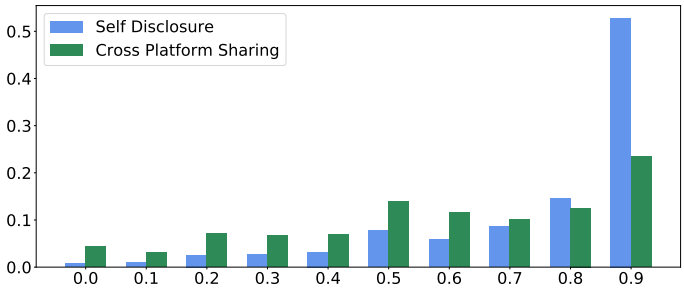
- Edit Similarity: This metric is similar to normalized Levenshtein distance but instead of dividing by the length of the shorter username, the Levenshtein distance is divided by the length of the longer username. The edit similarity between *rishabhk* and *rishabh* is 0.125 whereas it is 0.75 between *rishabhk* and *iiit.pk*. Lower edit similarity means it will take less time to make the strings equal, so they are more similar.

- Keyboard Typing Distance: The approximate distance traversed on a standard QWERTY keyboard while typing out the username. This metric is obtained by calculating the average euclidean distance between each character in the username with row and column of the key serving as its coordinates. For two given strings, we take absolute difference of the average euclidean typing distance to type characters in the input strings. For instance, the keyboard typing distance between *rishabh* and *iiit.rkaushal* is 0.49 and it is 1.8 between *rishabh* and *iiit.pk*. Lower the keyboard typing distance, more likely the strings are similar.

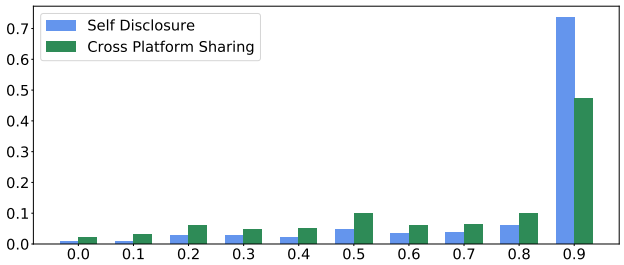
- LCS Similarity: Given two strings (display names) this metric is defined as the ratio of the length of the longest common string to the minimum length among the two strings. Its value lies between 0 and 1 and a greater value indicates a higher degree of similarity between the two display names. For instance, the LCS similarity between *rishabh* and *iiit.rkaushal* is 0.375 and it is 0 between *rishabh* and *pk*. Higher the LCS similarity, more likely the two strings are similar.

4.5.3 Behavioral Bias Characterization

To detect the existence of user behavioral biases in CPS and SD datasets, we study the distribution of lexical similarity features (Jaccard similarity and edit distance) measured on user behaviors in terms of their configuration of the username and display name. From the distribution of Jaccard similarity values for usernames in linked identities obtained from CPS and SD in Figure 4.5(a),



(a) Jaccard Similarity (JS) on User Names. 50% of user identity pairs obtained from self-disclosure have JS value in their usernames 0.9 as opposed to only 23% from cross-posting.



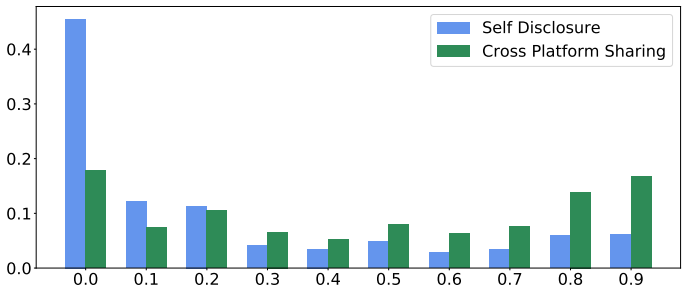
(b) Jaccard Similarity (JS) on Display Names. More than 70% of user identity pairs from self-disclosure have JS in display names value 0.9 as opposed to only 43% from cross-posting.

Figure 4.5: Distributions of Lexical Features (Jaccard Similarity) which shows that usernames and display names of user identity pair obtained using self-disclosure method exhibit higher lexical closeness than those obtained using cross-posting.

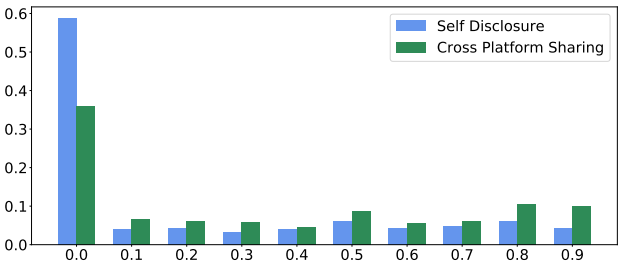
it is evidently clear that almost 50% of linked identities in SD dataset have Jaccard similarity on usernames value greater than or equal to 0.9 as compared to only 21% in the CPS dataset. This clearly shows that users who self-disclose their usernames are *lexically more similar* than those who cross-post. We perform KS-Test for two samples, and it turns out that KS-statistic is 0.33 with p-value less than 0.05 which is indicative that the two distributions are different from each other.

A similar trend is observed in the other user behavior of configuring display names. Here as well as shown in Figure 4.5(b), over 70% of SD dataset user identities have Jaccard similarity on display names greater than 0.9 as opposed to only 45% in CPS dataset. KS-Test performed on the two distributions gives KS-statistic of 0.2 and p-value less than 0.05 which means that distributions are different from each other. When we change the measurement metric from Jaccard similarity to edit distance, the trend of usernames (Figure 4.6(a)) and display names (Figure 4.6(b)) coming from SD dataset exhibiting higher lexical similarity continues. KS-Test on edit distance distributions coming from CPS and SD dataset indicates that they are different distributions. This clearly

shows evidence for the existence of user behavioral biases in terms of configuring their usernames and display names.



(a) Edit Distance (ED) on User Names. 45% usernames of user identity pair from self-disclosure have 0.0 ED than only 18% from those obtained from cross-posting.

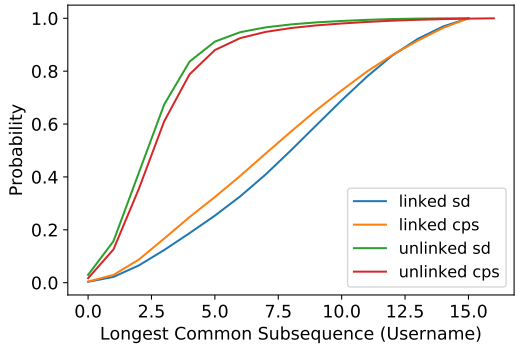


(b) Edit Distance (ED) on Display Names. 57% display names of user identity pairs obtained through self-disclosure have 0.0 ED as compared to 35% from cross-posting.

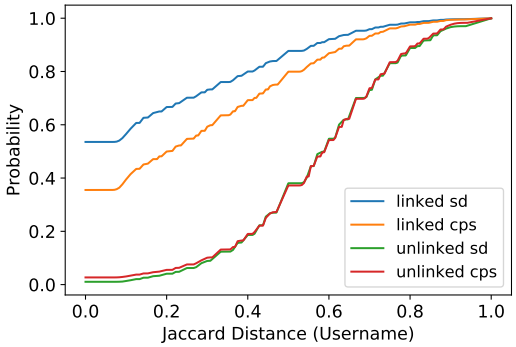
Figure 4.6: Distributions of Lexical Features (Edit Distance) which reaffirms that usernames and display names of user identity pair obtained using self-disclosure method exhibit higher lexical similarity than those obtained using cross-posting.

Next, we study the cumulative distribution of feature values for both linked and unlinked user identities obtained from cross-posting, self-disclosure, and negative sampling. We first study the length of the longest common subsequence (LCS) in usernames at character-level. As depicted in Figure 4.7(a), most (90%) of the unlinked user identities have an LCS of length less than 6. More proportion of linked user identities in the CPS dataset have higher LCS length than those from the SD dataset. In terms of distance metrics, namely Jaccard distance on usernames (Figure 4.7(b)) and normalized Leveinshtein distance on usernames (Figure 4.7(c)), we observe that unlinked user identities are more distant than linked identity pairs. Given that both are distance variants, so the proportion of linked identities having a higher distance in their usernames and display names is less.

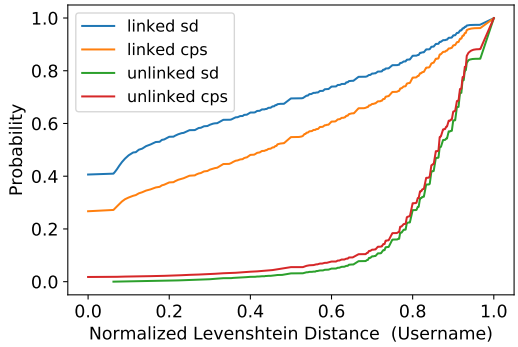
However, a significant gap between the blue and orange curves depicting linked identities from SD and CPS datasets clearly indicates the presence of behavioral biases in these datasets. We say



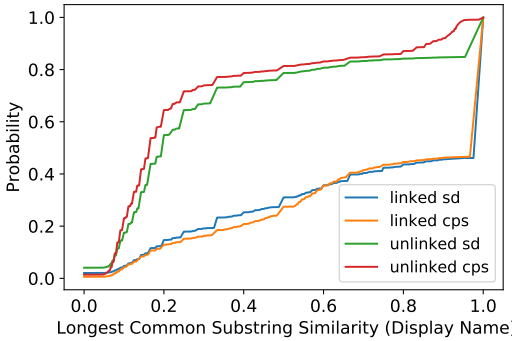
(a) CDF of Longest Common Subsequence on Usernames. 90% of unlinked user identities have LCS of a length less than 6.



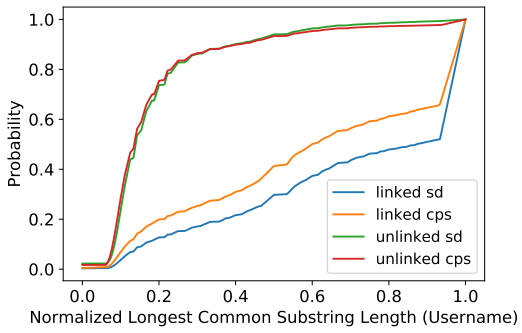
(b) CDF of Jaccard Distance on Usernames. Linked identity pairs are more closer (lessor distance) than unlinked identity pairs.



(c) CDF of Normalized Leveinshtein Distance on Usernames. Unliked identity pairs are evidently more lexically distant than linked identity pairs.



(d) CDF of LCS Similarity on Display Names. 80% of unlinked user identity pairs have LCS less than 0.3 which shows lower similarity.



(e) CDF of Normalized LCS on Usernames. 90% of unlinked user identity pairs have normalized LCS around 0.2 or less which shows lower similarity.

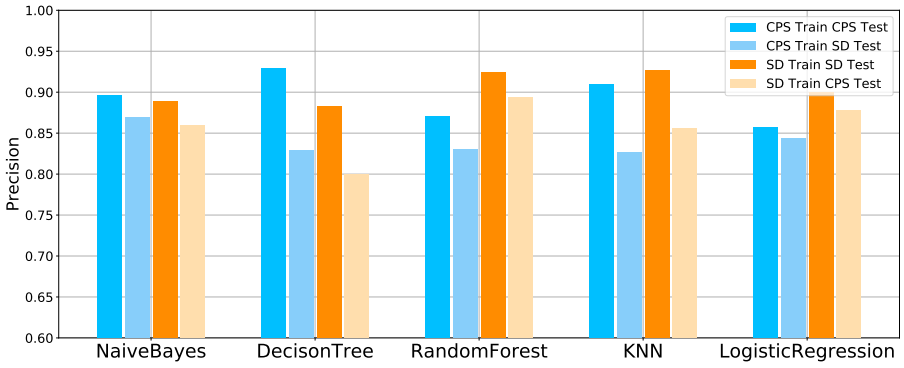
Figure 4.7: Cumulative Frequency Distribution (CDF) plots of Lexical Features on User Names and Display Names obtained from CPS and SD Datasets.

significant because when two sample S-K test was performed on these two distributions (blue and orange), then the p-value turned out to be less than 0.01 at a significance level of α=0.05 with large D-statistic. This proves that distributions of Jaccard distance and normalized Levenshtein distance for linked user identities in SD and CPS datasets are drawn from different distributions, consequently, it establishes behavioral biases manifested through these lexical similarity metrics. In the case of LCS similarity feature for display name (Figure 4.7(d)) and normalized LCS on usernames (Figure 4.7(e)), we observe that linked user identities in SD and CPS datasets exhibit more similarities than unlinked user identities. Also we observe in Figure 4.7(e) that among the linked user identities, the normalized length of LCS on username is smaller in CPS dataset than SD dataset. These CDF plots provide an evidence of behavioral biases in users who cross-post and self-disclose which is manifested in the form of these lexical features.

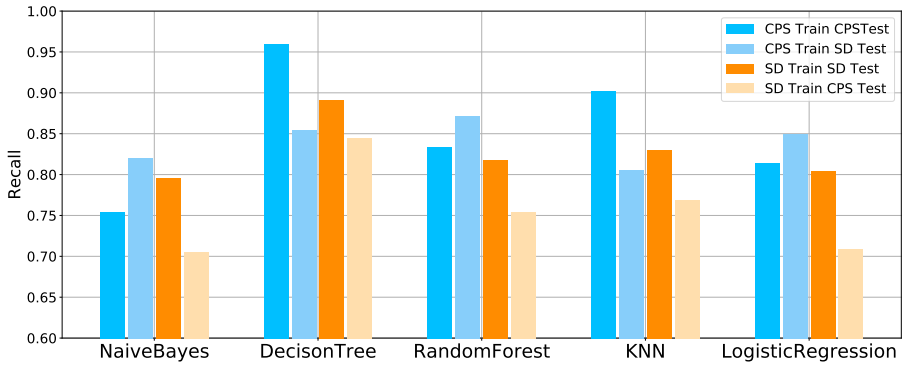
4.6 Impact of Behavioral Biases on Identity Linkage Models

In this section, we propose a methodology to study the impact of behavioral biases in classification models for identity linkage. After having proven the presence of behavioral biases and characterized them, we measure their impact on the decision making capability of identity linkage models. Recall from equation 4.1 that we solve the identity linkage problem by constructing a classification model. A robust classification model ought to be generalizable, in other words, it is expected to perform equally well irrespective of the source of training data. To test the robustness of identity linkage models built in the presence of behavioral biases that are manifested in lexical features derived from usernames and display names, we design four experiments, based on the idea of cross-dataset generalization analysis [150, 151]. In first and second, we train the model using the same CPS dataset, but perform testing using CPS dataset & SD dataset, respectively. In third and fourth, we train the model using the same the SD dataset and test it using the SD dataset & CPS dataset, respectively. We consider five classification algorithms namely Naive Bayes, Decision Tree, Random Forest, KNN, and Logistic Regression in our experiments.

Figure 4.8(a) depicts the precision of these algorithms for all four experimental scenarios in predicting a linked identity pair to be correctly belonging to the same individual. As evident from Figure 4.8(a), irrespective of the learning algorithm adopted, the precision of models trained on the CPS dataset & tested on the CPS dataset is better than those tested on the SD dataset. Similarly, the precision of models trained on the SD dataset & tested on the SD dataset is far better than those tested on the CPS dataset. This proves that the classification models get significantly biased with the dataset used to perform training. This could only happen when there are biases that exist in the dataset. In the case of recall, depicted in Figure 4.8(b), a similar trend is observed in the case of models trained on the SD dataset & tested on the SD dataset outperforms those tested on the CPS dataset. However, no conclusive trend is obtained in the case when models are



(a) Precision Values of CPS and SD driven models



(b) Recall Values of CPS and SD driven models

Figure 4.8: Impact of Behavioral Biases in CPS and SD dataset on performance (precision and recall) of Classification Models. Performance of cross-dataset models is poorer as compared to models trained and tested on same dataset.

trained on CPS dataset & tested on CPS and SD dataset, separately. This is due to the fact that linked identities in the CPS dataset exhibit lower lexical similarities in the features than the SD dataset. Consequently, when models are trained on the CPS dataset, the training dataset is unable to provide the necessary discriminative training required for the model to be decisive.

4.7 Quantification of Bias

After detecting behavioral biases in user identities, characterizing them, and measuring their impact on identity linkage models, we propose a *novel* approach that quantifies biases by leveraging from a well-established discrimination measurement approach namely *situational testing* [99]. Before explaining our approach, we briefly explain the concept of situational testing from the perspective of discrimination studies.

Situational Testing

In the context of discrimination studies, we refer a specific group of users as a *protected group* based on values of one or more *protected attributes* (like gender, race, locality, etc.) and the goal is to *protect* this group from discrimination based on protected attributes. As per situational testing, a data record (representing a user in the real world) is considered to be *discriminated* if a significant difference is observed in its treatment (prediction of a label in case of learning model-based decision making) with respect of its neighbors in protected group and neighbors not in the protected group. For illustration, consider a job suitable for both males and females, in which both males and females apply. And the job application process involves a stage in which a learning model-based applicant screening is adopted. Given that learning model is to be trained on historical decisions, so the biases (if any) that exist in the training data (in this case, say more males were offered a job in the past), are going to impact the learning model, make the decision outcomes of the model biased as well. Situational testing is an approach that quantifies such biases by leveraging K-Nearest Neighbor (KNN) classification technique. More formally, the following steps are performed in situational testing.

1. Consider a dataset D of decision records, having n data instances $d_1, d_2, ...d_n$ and the class attribute represented by $class(d_i)$. In the above example, $class(d_i)$ can be either *accept* or $reject$ job application.

2. Consider a single protected attribute represented by *proc* which takes on categorical values. In the above example, *proc* attribute is *gender* taking on two values *male* and $female$ and the protected group ($P(D)$) is all females.

$$P(D) = d_i : proc(d_i) = female, \forall i = 1...n \tag{4.2}$$

   Similarly, the unprotected group ($UP(D)$) becomes.

$$UP(D) = d_j : proc(d_j) = male, \forall j = 1...n \tag{4.3}$$

3. Take a suitable distance function as required in KNN algorithm as f_{dist}, but define it only for non-protected attributes.

4. For each test record d_{test}, find its K-nearest neighbors using f_{dist} in both protected group $P(D)$ and unprotected group $UP(D)$ in the training data. Accordingly, we define two variables for each d_{test} as below.
   p_1: proportion of records in $P(D)$ with same decision as d_{test}
   p_2: proportion of records in $UP(D)$ with same decision as d_{test}

5. Lastly, we define $t = p_1$ - p_2, $-1 \leq t \leq 1$ and find the distribution of values of t which indicates the amount of discrimination with which each d_{test} gets affected. If $t = 0$, there is no discrimination but higher the values of t towards either -1 or $+1$, more is the severity with which the test record d_{test} is affected.

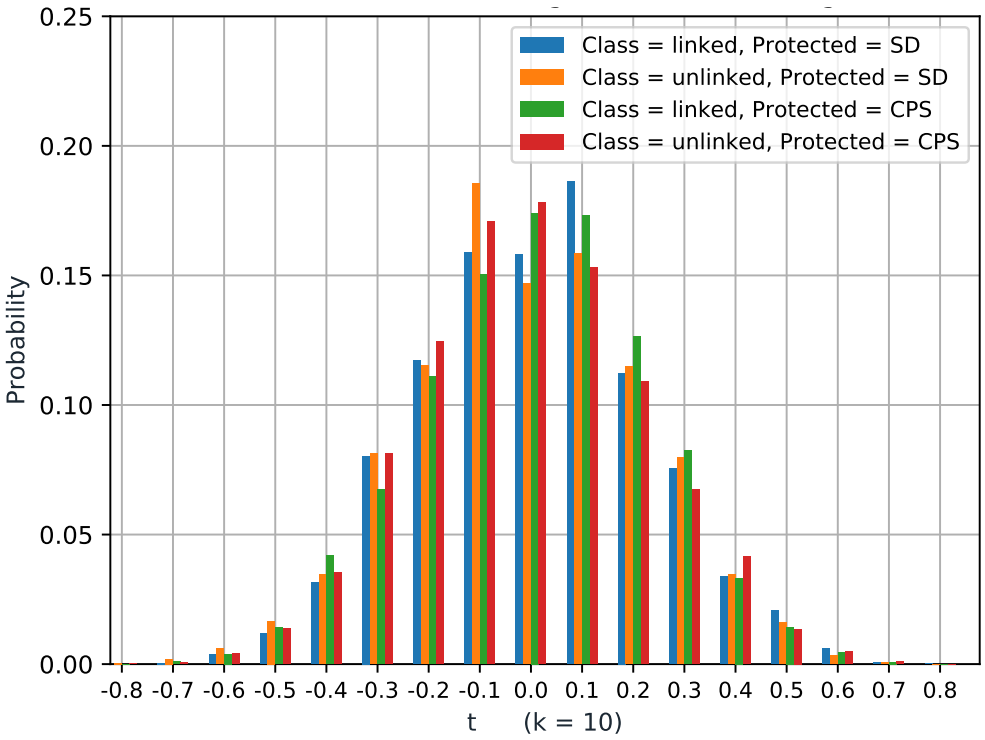


Figure 4.9: Effect of Biases on Linked and Unlinked User Identities in both scenarios when CPS and SD was taken protected group separately. Probability distributions of $t - values$ are spread on both positive ($t > 0$) and negative ($t < 0$) sides which indicates that behavioral biases affect many test records.

Our Proposed Approach

In this section, we discuss how we *apply* situational testing in the context of our problem of quantifying biases. To the best of our knowledge, this is the first work which is leveraging situational testing, which was originally proposed as a measurement methodology to study discrimination, to quantify biases. To adopt situational testing, we propose the following in our design.

1. Create a combined dataset D which is drawn from two datasets of linked user identities namely D_{CPS} and D_{SD} obtained by leveraging cross-posting and self-disclosure user behavior, respectively. Alongside the linked user identities, we also take the unlinked user identities as per the negative sample generation procedure explained earlier.

2. While combining, create a new attribute *data_source* which would take values CPS or SD, and consider this new attribute as a *protected attribute* in order to apply discrimination measures in general and in particular situational testing.

3. Perform two sets of experiments, first by treating records containing $data\_source = SD$ as a protected group, and second by treating records containing $data\_source = CPS$ as a protected group.

4. The decision to be taken in the context of identity linkage problem is whether the user identity pair belong to the same individual, referred as *linked* or different individuals referred to as *unlinked*. In all our experiments, we focus our attention on the decision of *linked*, unless otherwise stated.

Using the above approach, we are able to apply the concept of situational testing to study the quantification of behavioral biases in user identities. Next, we explain the results for different experimental designs.

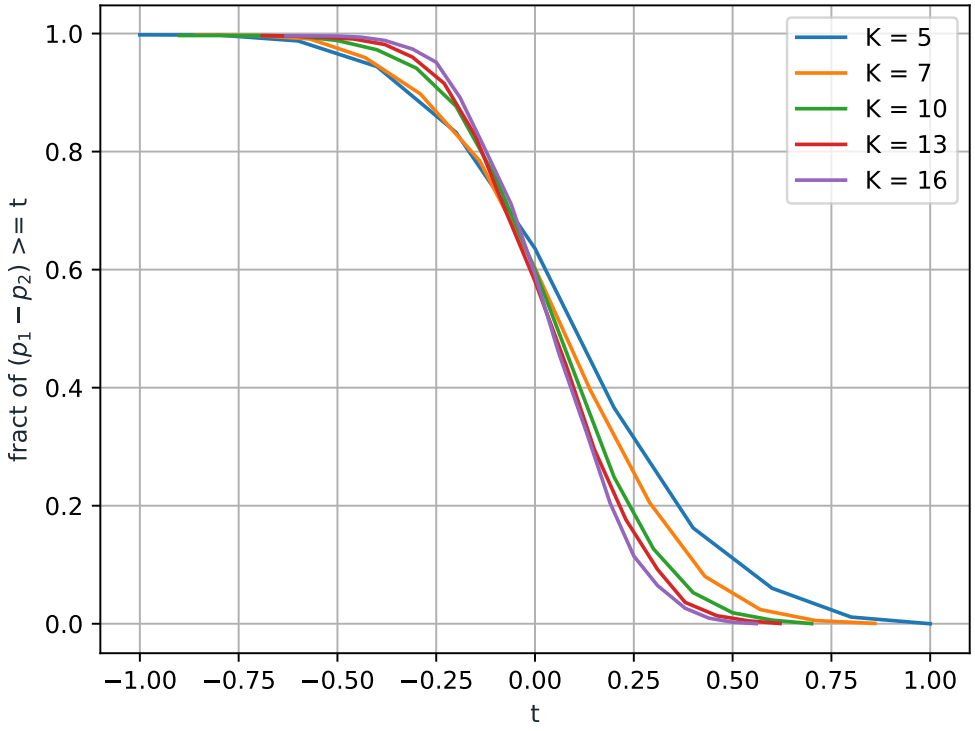


Figure 4.10: Effect of K-Nearest Neighbors on Biases on Linked User Identities when SD was taken protected group. We observe that as the value of K increase, the amount of biases measured through the cumulative distribution of $p_1 - p_2$ values decrease.

Results

We design experiments to answer three questions, (1) *Are both decision classes (linked and unlinked) equally affected by biases?* To address this, we measure the impact of biases using the situational testing framework (keeping K=10 in KNN algorithm). We plot (Figure 4.9) probability distributions of $t-values$, on both class values namely *linked* and *unlinked* user identities and in both scenarios where $data\_source = SD$ and $data\_source = CPS$ are taken as a protected group, separately. It is clearly evident that probability distributions of $t-values$ are spread on both positive ($t > 0$) and negative ($t < 0$) sides which indicates that behavioral biases affect many test records.

To measure which scenario is most affected by the biases, we find create two metrics t_{sum} and t_{wt-sum} as defined below.

$$t_{sum} = \sum_{t=0.1}^{t=1.0} prob(t) + \sum_{t=-0.1}^{t=-1.0} prob(t) \tag{4.4}$$

$$t_{wt-sum} = \sum_{t=0.1}^{t=1.0} t \times prob(t) + \sum_{t=-0.1}^{t=-1.0} t \times prob(t) \tag{4.5}$$

After computing t_{sum} and t_{wt-sum} for all four scenarios, it turns out that t_{sum} is highest (only 1% higher) for *unlinked* class when $data\_source = CPS$ was taken as a protected group and t_{wt-sum} is highest (only 0.2% higher) for *unlinked* class when $data\_source = SD$ was taken as a protected group.

(2) *What is the effect of the number of nearest neighbors (K in K-NN algorithm) on the severity of biases?* To understand the impact that K-Nearest Neighbors as per the KNN algorithm have on the extent of biases, we plot (Figure 4.10) cumulative distribution of $p_1 - p_2$ values for different values of K. From Figure 4.10, it is observed that as the value of K increase, the amount of biases measured through the cumulative distribution of $p_1 - p_2$ values decrease. This trend is consistent with every increase in the value of K. The intuitive explanation for this observation is that as we increase the value of K, we increase the probability of obtaining instances in training which belongs to both *linked* and *unlinked* classes.

(3) *Does the amount of training has an impact on the amount of biases suffered by test data instances?* We plot (Figure 4.11) the amount of biases suffered by test instances through probability distribution of $t-values$ for the varying amount of training data size, keeping the value of $K = 10$ and treating SD dataset ($data\_source = SD$) as a protected group. Looking at Figure 4.11, one concludes that biases exist across all scenarios irrespective of the amount of training dataset and the $t-values$ distribution follows a *normal distribution*. This rejects the intuitive notion that a large amount of training would nullify the effect of biases, and hence there is a need for a methodology to mitigate the effect of biases.

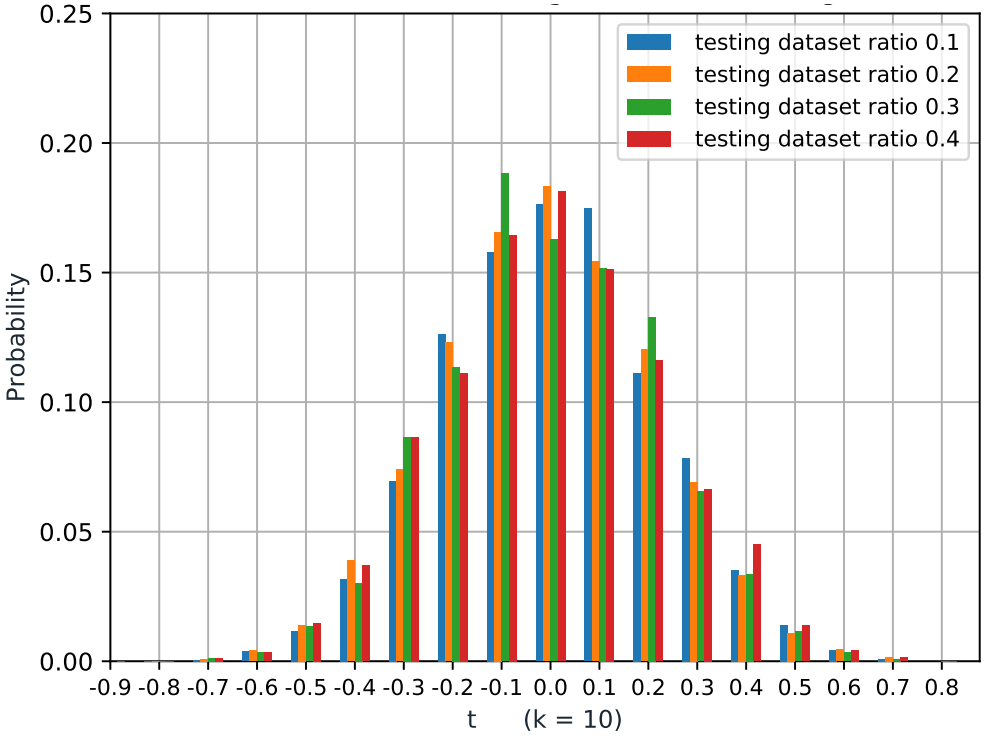


Figure 4.11: Effect of Biases on Linked User Identities with varying Training Sizes, considering SD as protected group. Biases exist across all scenarios irrespective of the amount of training dataset.

4.8 Discussions, Limitations & Conclusions

In this section, we first explain the key observations and takeaways from our work in this Chapter.
User Behavioral Characterization: In regard to the presence of behavioral biases in user identities datasets, we find that users who indulge in self-disclosure intend to keep their usernames and display names more lexical similar as compared to those who cross-post. Behaviorally speaking, we find that users who disclose [97] by mentioning details of their identities on other social networks in the bio-field of their Twitter identity, make their identities appear similar and hence, their usernames & display names exhibit high lexical similarity across social networks. A plausible explanation for this behavior could be that they are conscious about making the identities look similar. In contrast, the users who cross-post, typically do it occasionally and we believe that they are not likely to be conscious to explicitly keep their identities similar, therefore their usernames & display names exhibit lessor lexical similarity. More experiments need to be conducted to ascertain the exact reasons for these user behaviors. From the privacy standpoint, it would be good to have a system that nudges [72] such users about linkability of their identities across multiple social networks so that they can make an informed decision about cross-posting.
Adoption of Best Practices: Further, it has been observed that researchers solving the problem

of identity linkage have mostly relied upon a single data source to evaluate their proposed models. Therefore, a key takeaway from our work would be to re-evaluate the prior works in the light of biases that could be manifested in their dataset. Detection and neutralizing biases in user identity datasets need to become an essential pre-processing step before going ahead in evaluating proposed solutions.

Application of Discrimination Measures: Through this work, we have proposed an effective strategy to leverage discrimination measurement metrics to detect, quantify, and mitigate biases in the dataset which are collected relying upon human behaviors. Just as we have leveraged a discrimination measurement framework by considering *data_source* attribute as a protected attribute, we believe that works in solving problems in other domains through data-driven approaches would also benefit similarly.

Limitations in this chapter can be observed at two levels. At the *first* level, the fact that we study user behavior only in terms of usernames and display names, can be further extended to other profile attributes and also other behaviors in terms of content posting and networks that users keep. Given the increasing restrictiveness in the social network APIs, obtaining information about content and network would be a challenge, nevertheless. At the *second* level, the methodology for detecting, quantifying and preventing biases can be further strengthened by drawing more ideas from bias studies [44, 125], discrimination studies [52, 131] and fairness preserving algorithmic studies [20, 86, 174, 177].

Finally, in the context of the three research questions in this chapter, we conclude that behavioral biases exist in user identity datasets obtained by leveraging cross-posting and self-disclosure. We study two user behaviors namely username and display name configuration, and find biases are present which get manifested in the form of lexical similarity features. Identity linkage models are affected by 5-20% when trained on data collected using the cross-posting method and tested on data obtained from self-disclosure and vice-versa. We quantify the extent of damage caused by these biases in terms of the number of biased decisions (15-20%) made by the classification models.

Chapter 5

NeXLink: Node Embedding Framework for User Identity Linkage

So far, we have discussed different methods for the collection of linked user identities and biases that exist in the datasets of linked user identities, thus obtained. In this chapter[1], we present a node embedding based approach for the detection of linked user identities. In the context of our proposed node embedding based approach, we refer to user identity pairs belonging to the same person as Cross-Network Linkages (CNLs). We model the social network as a graph where user represents node and friend relation represents edge. We explore the question, *whether we can obtain effective social network graph representation such that node embeddings of users belonging to CNLs are closer in embedding space than other nodes, using only the network information.* To this end, we propose a modular and flexible node embedding framework, referred to as *NeXLink*, which comprises of three steps. First, we obtain local node embeddings by preserving the local structure of nodes within the same social network. Second, we learn the global node embeddings by preserving the global structure, which is present in the form of common friendship exhibited by nodes involved in CNLs across social networks. Third, we combine the local and global node embeddings, which preserve local and global structures to facilitate the detection of CNLs across social networks. We evaluate our proposed framework on an augmented (synthetically generated) dataset of 63,713 nodes and 817,090 edges and a real-world dataset of 3,338 Twitter-Foursquare node pairs. Our approach achieves an average hit rate of 98% and 88% in augmented and real-word dataset, respectively, for detecting CNLs across social networks and significantly outperforms previous state-of-the-art methods.

[1]Mostly taken from our published paper. **Rishabh Kaushal**, Shubham Singh, and Ponnurangam Kumaraguru. NeXLink: Node Embedding Framework for Cross-Network Linkages Across Social Networks. In *Proceedings of International Conference on Network Science, 2020.*

5.1 Introduction

Users join multiple online social networks (OSNs), and in such a scenario, we refer to the user identities across multiple OSNs belonging to the same individual, as *cross-network linkages* (CNLs) in this chapter. The conventional approach, as we discuss in Section 2.3 of Chapter 2, is to *recast* the problem of identity linkage as a machine learning based classification problem and construct hand crafted features from user profile [88, 122, 165, 175], user content [23, 41] and user's friend network [80, 194]. This requires meticulous formulation and computation of features to build accurate models, which is quite challenging. With the recent advancements in network embedding [30] and deep learning [32], as we discuss in Section 2.4 in Chapter 2, the objective is to find effective graph representations that provide an alternate direction to solve the problem.

In this chapter, we propose a solution based on the construction of effective graph representations. The goal is to learn node embeddings in a social graph such that nodes with similar characteristics are represented by similar node embedding vectors. In the context of our problem, we ask the question *whether we can obtain effective social network graph representation such that node embeddings of users belonging to CNLs are closer in embedding space than other nodes.* As we depict in Figure 5.1, the goal is to propose an embedding framework that transforms nodes into embedding vectors such that nodes present in linked identities are closer in embedding space than other nodes. To this end, we propose a three-step *NeXLink* framework that learns node representations to detect CNLs across social networks. In the first step, we preserve the local structure of nodes within the same network. In social networks, these local structures would comprise of friendship relation or follow-followee relation maintained by user identities. In particular, we learn node embeddings of nodes within the same network using the normalized edge weights so that nodes that are structurally near to each other, their corresponding embeddings are also close in embedding space. In

(a) (b)

Figure 5.1: Our proposed NeXLink framework learns node embeddings from two social networks (represented as graphs, on the left side) with few cross-network linkages. On the right side, we depict embedding space in which nodes corresponding to user identities belonging to same individual are closer than other nodes.

the second step, we preserve the global structure of nodes connected across multiple networks. In social networks, these global structures would comprise cross-platform linkages representing user identities across social networks belonging to the same individual. We expect these linked identities to exhibit a number of common friends across social networks. In particular, we learn the node embeddings of nodes that are part of Cross-Network Linkages (CNLs) by biasing the random walk in proportion to the common friendship. As a result, node embeddings of nodes that are part of CNLs with more common friends are closer in embedding space. In the third step, we directly leverage the node embeddings to evaluate their efficacy in the detection of cross-network linkages across social networks.

We evaluate our proposed approach of the NeXLink framework on two datasets. The first dataset is an augmented dataset synthetically generated using the Facebook social network [154] comprising of 63,713 nodes (users) and 817,090 edges. Our approach works well in all possible augmentations of the Facebook dataset achieving an average Hit@1 rate of 98%, which means that the probability of hitting on the correct cross-network linkage at rank-1 among the top-k retrieved candidates is 98%. Further, our approach outperforms the state-of-the-art prior approaches of node representations, namely LINE [148] and DeepWalk [124] on synthetically generated graphs, which we refer to as augmented dataset. The second dataset comprises of a real-world dataset of Twitter-Foursquare social networks [184] comprising of 3,338 nodes (user) pairs. We find that, except for the Hit@1 rate, our approach works better than the state-of-the-art prior approaches of user identity linkages, namely IONE [94] and REGAL [56] at Hit@5 and above. The key contributions of our work in this chapter are as below.

- We propose a modular and flexible NeXLink framework as a two-step optimization process that preserves local structure within the same network and preserves global structure manifested in the form of cross-network linkages.
- We extensively evaluate our framework on two datasets, one augmented dataset is obtained from Facebook and other real-world datasets comprising Twitter-Foursquare node pairs. Our framework works well on the synthetically generated dataset and outperforms prior node representation approaches (LINE and DeepWalk) and identity linkage approaches (IONE and REGAL).

5.2 Related Work

There are two broad approaches to address the problem of user identity linkage, which we categorize as the machine learning based approach and network embedding approach.

5.2.1 Machine Learning Approach

The machine learning approach, as we discuss in Section 2.3 of Chapter 2, relies upon the features derived from users' behaviors within and across social networks. Prior works consider three types of users' features derived from profile, content, and network maintained by users. *Profile-based features*: The *first* type comprises of the attributes that users configure on their profile page that are made visible and their similarity across social networks. Perito et al. [122] investigated the likelihood of user identities belonging to the same person if usernames on these identities are similar. Zafarani et al. [175] proposed a methodology (MOBIUS) that creates features derived from prior usernames used by individuals across multiple social networks. Li et al. [88] explained the patterns and similarities of display names configured by users across social networks. *Content-based features*: The *second* type of users' behavior comprise of content posted by users in online social networks. Prior works derive features from the content and meta-data related to content. Gogo et al. [41] leveraged geo-tagged location, timestamp, and writing style in posts made by users as features to logistic regression classifier. Chen et al. [23] extracted users' spatial features using density-based clustering and temporal features using a Gaussian Mixture Model. *Network-based features*: The *third* type of users' behavior comprises of the friends (or followees-followers) maintained by users. Korula et al. [80] presented a theoretical formulation of the problem of finding cross-network linkages on network models based on random and preferential attachment models. Zhou et al. [194] proposed an approach that they refer to as Friend Relationship based User Identification algorithm without Prior knowledge (FRUI-P), which is an unsupervised approach that extracts features based on friend network maintained by users.

5.2.2 Network Embedding Approach

Recently, there are a few prior works that leverage the network embedding approach, which we discuss in Section 2.4 of Chapter 2, whose aim is to learn a low dimensional representation for a given node in a graph. We categorize these prior methods in the field of network embedding into two main categories, as explained below.

Problem independent approaches: These works only aim to learn generic low-dimensional representations without focusing on user linkage problems. The objective is to learn effective node representations in low dimensions. Tang et al. [148] proposed a framework for network embedding in large graphs to preserve node structures of nodes that are directly connected (first-order node proximity) and connected at a distance of two hops (second-order node proximity). Perozzi et al. [124] leveraged the notion of the skip-gram model in language modeling to perform truncated random walks in order to learn latent representations of nodes in a graph. Wang et al. [158] preserved the first and second-order node proximity using a semi-supervised deep learning model. Grover et al. [47] extended the notion of a random walk by introducing biased walks in node neigh-

borhood to learn feature representations of the node in a network. Xu et al. [168] proposed two embeddings for each node that capture the structural proximity of nodes as well as the semantic similarity, which they express in terms of common interests. Liang et al. [89] presented a dynamic user and word embedding model (DUWE) that monitors over some time, the relationship between user and words to model their embeddings. Liu et al. [92] explained a self-translation network embedding (STNE) framework that is a sequence-to-sequence framework taking into consideration both content and network features of the node.

Problem dependent approaches: These learn low-dimensional embedding focusing on a specific problem, which in our case is to detect cross-network linkages representing user identities across social networks. Liu et al. [94] proposed an input-output node embedding (IONE) framework to align user identities across social networks belonging to the same person by learning node representations that preserve follower-followee relationships. Man et al. [102] introduced a framework referred to as PALE, which predicts anchor links via embeddings. First, it converts a social network into a low dimensional node representation. They follow it up by learning a matching function that is supervised by known anchor links. Heimann et al. [56] explained the REGAL framework, which stands for representation learning-based graph alignment based on the cross-network matrix factorization method. Wang et al. [164] proposed LHNE mode referred to as linked heterogeneous network embedding, which creates a unified framework to leverage structure and content posted by users for learning node representations. Xie et al. [167] used the concept of factoid embedding, which is an unsupervised approach to perform user identity linkage. Our proposed approach outperforms some of these existing approaches, as we explain later in this chapter.

5.3 Proposed Approach

In this section, we discuss our proposed NeXLink framework for effective representation and detection of cross-network linkages across social networks. We consider two social networks X and Y as two undirected graphs $G_X(V_X, E_X)$ and $G_Y(V_Y, E_Y)$, where V_X & V_Y represent the nodes (users) of graphs and E_X & E_Y represent the edges. An edge between nodes u_i and u_j indicates friendship relation between users u_i and u_j. We divide the set of node pairs (u_i^X, u_j^Y) across social networks X and Y into two types, namely, cross-network linkages, denoted by $CNL(V_X, V_Y)$ and other pairs are denoted by $NCNL(V_X, V_Y)$. Nodes u_i^X and u_j^Y belonging to social networks X and Y are referred to as cross-network linkage if u_i^X and u_j^Y belong to the same individual and the pair $(u_i^X, u_j^Y) \in CNL(V_X, V_Y)$ else $(u_i^X, u_j^Y) \in NCNL(V_X, V_Y)$. Further, it may be observed in Figure 5.2, that the two users represented as two nodes u_i^X and u_j^Y have a^X, b^X and c^X as friends in social network X and same friends a^Y, b^Y and c^Y in social network Y. We refer to such familiar friends as common friendship and leverage this behavior in learning node representations in our NeXLink framework. Besides familiar friends, each node also has some friends who are specific to one social

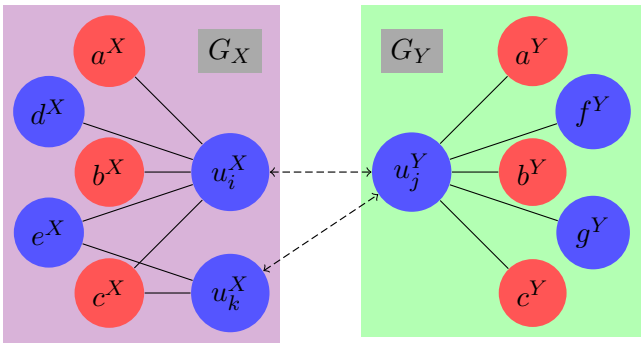


Figure 5.2: Illustration of common neighbors of user identities u_i^X and u_j^Y belonging to networks G_X and G_Y. Since all neighbors are common, it is highly likely that u_i^X and u_j^Y belong to same individual than u_k^X and u_j^Y.

network only. In Figure 5.2, nodes d^X and e^X are friends of u_i^X in only social network X whereas nodes f^Y and g^Y are friends of u_j^Y in only social network Y. We note that the above formulations for undirected graphs are also applicable in directed graphs, in which case the friendship relation is replaced by follow-followee relation using directed edges.

5.3.1 Problem Statement

Given two graphs $G_X(V_X, E_X)$ and $G_Y(V_Y, E_Y)$ as input, we define cross-network linkage as the set of user identity pairs across these two networks X and Y, denoted by $CNL(G_X, G_Y)$, which belong to the same person. Similarly, we denote all other user pairs which do not belong to the same person by $NCNL(G_X, G_Y)$. The goal of network embedding function (denoted by f_{emb}) is to transform each user identity $u_i^X \in V_X$ and $u_j^Y \in V_Y$ into low *d-dimensional* vectors z_i^X and z_j^Y such that if user identities u_i^X and u_j^Y belong to the same individual (i.e. they represent cross-network linkage), then their corresponding node embeddings z_i^X and z_j^Y are closer in embedding space else they are far apart.

$$
\begin{aligned}
z_i^X &= f_{emb}(u_i^X), \forall u_i^X \in V_X. \\
z_j^Y &= f_{emb}(u_j^Y), \forall u_j^Y \in V_Y. \\
\textit{such that} & \\
sim(z_i^X, z_j^Y) &>> sim(z_k^X, z_j^Y) \textit{ and} \\
\exists\ (u_i^X, u_j^Y) \in CNL(V_X, V_Y) &\wedge (u_k^X, u_j^Y) \in NCNL(V_X, V_Y).
\end{aligned}
\tag{5.1}
$$

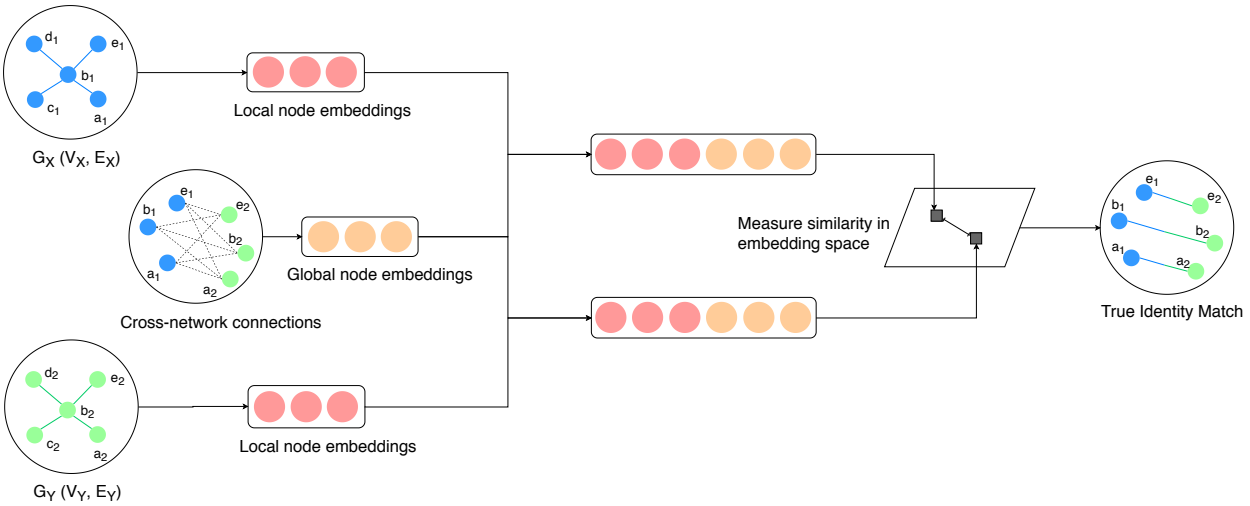


Figure 5.3: NeXLink Framework. Architecture diagram of our proposed framework that learns node embeddings from two social networks (represented as graphs) to represent the cross-network linkages across social networks. Local node embeddings are concatenated with global node embeddings to generate final node embeddings.

5.3.2 NeXLink Framework

The goal of our proposed NeXLink node embedding framework is to obtain representations of nodes in two networks X and Y such that node pairs participating in cross-network linkages have similar node embeddings than other node pairs. To achieve this goal, we follow a two-step approach, as depicted in Figure 5.3. In the first step, structural similarities of nodes within or local in their respective networks are preserved independently of the other network. In the second step, similarities of nodes across (or global) the two networks are preserved using a common friendship relationship. Given the two-step process, the embedding function f_{emb} can be broken down into two embedding functions, as shown below.

$$\begin{aligned} z_i^X &= f_{global}(u_i^X) \oplus f_{local}(u_i^X), \forall u_i^X \in V_X. \\ z_j^Y &= f_{global}(u_j^Y) \oplus f_{local}(u_j^Y), \forall u_j^Y \in V_Y. \end{aligned} \tag{5.2}$$

There can be different ways of combining global and local node embeddings. However, it turned out that concatenation is the best operation $\oplus$ to combine local and global node embeddings, which we finally used in our proposed NeXLink framework. Further, we note that our proposed approach makes use of only the network structure in the two social networks. However, it can be easily extended to include other sources of information from content and profile information of users, which we leave for future work.

Step 1 - Preserving Local Structure Within Social Networks We perform the first step

on the intuition that directly connected user nodes within their respective social networks are likely to exhibit similar characteristics, based on the well established social behavioral principle of homophily [107]. Given two nodes u_i^X and u_k^X in same social network X, the goal is to define an encoding function f_{local} that takes these nodes as input and learns their *d-dimensional* embedding vectors $z_i^X \in R^d$ and $z_k^X \in R^d$, respectively as described below.

$$\begin{aligned} z_i^X &= f_{local}(u_i^X) \\ z_k^Y &= f_{local}(u_k^X) \end{aligned} \tag{5.3}$$

To learn z_i^X and z_k^X for all nodes in V_X, we rely upon the probabilistic approach. The empirical probability of the relationship between two nodes u_i^X and u_k^X within the same social network X can be defined as the normalized weight of edge ($w_{i,k}^X$) between the nodes. Since we consider only the structural information of the network, therefore, for this work, we consider weights to have binary values 1 or 0, depending upon whether there is an edge or not, respectively. In general, the weight of the edge between nodes is intuitively proportional to the similarity between two nodes. Similarly, we can measure other criteria like content similarity. However, we consider only the network structure similarity in this chapter. We employ a well-established network embedding algorithm, LINE [148], to preserve the local structure.

$$\hat{p}_{local}(u_i^X, u_k^X) = \frac{w_{i,k}^X}{\sum_{(i,j)\in E_X} w_{i,j}^X} \tag{5.4}$$

In the scenario when nodes u_i^X and u_k^X are represented by their embedding vectors z_i^X and z_k^X, respectively, the joint probability between two nodes u_i^X and u_k^X in same social network X can be expressed as below.

$$p_{local}(u_i^X, u_k^X) = \frac{exp((z_i^X)^T z_k^X)}{1 + exp((z_i^X)^T z_k^X)} \tag{5.5}$$

Optimization: To learn *effective* and *representative* node embeddings, the goal is to bring the estimated probability (equation 5.5) and empirical probability (equation 5.4) as close as possible. Taking into account all possible node pairs in a given social network, we obtain two probability distributions corresponding to estimated probability (equation 5.5) and empirical probability (equation 5.4). For the estimated probability distribution to be a good *approximation* of the empirical probability distribution, we take the help of the KL-divergence metric. KL-divergence measures the amount of information loss when another probability distribution approximates a given probability distribution. Hence, we use KL-divergence as the objective function (equation 5.6), and the goal

of optimization is to minimize the KL-divergence between these distributions.

$$\begin{aligned} O_{local} &= \sum_{(i,j)\in E_X} D_{KL}(\hat{p}_{local}||p_{local}) \\ &= -\sum_{(i,j)\in E_X} \hat{p}_{local} \times log\ (\frac{p_{local}}{\hat{p}_{local}}) \end{aligned} \tag{5.6}$$

This ensures that the learned embedding vectors preserve local structure among nodes within the same social network. In other words, the embedding vectors of nodes directly connected will be closer in embedding space. To make optimization as specified in equation 5.6 tractable, we follow the approach of negative sampling [111] which has been used in prior state-of-the-art LINE [148] algorithm for node embedding. Similarly, we can learn node embeddings in other social network Y. It may be noted that node embeddings for social networks X and Y are learned in this step *independently* of each other.

Step 2 - Preserving Global Structure Across Social Networks We propose the second step based on the intuition that user nodes with common friends (CF) across the social networks are likely to belong to the same person. The degree to which two nodes (users) u_i^X and u_j^Y on two social networks X and Y, respectively, having *common friendship*, is expressed as below.

$$CF(u_i^X, u_j^Y) = \frac{N(u_i^X) \cap N(u_j^Y)}{N(u_i^X) \cup N(u_j^Y)} \tag{5.7}$$

where $N(u_i^X)$ and $N(u_j^Y)$ represent the set of friends of i^{th} user in network X and j^{th} user in network Y, respectively. Higher is the value of common friendship (CF), more likely the users u_i^X and u_j^Y would belong to the same person. Therefore, the goal of second encoding function f_{global} is to take u_i^X and u_j^Y as inputs and generate *d-dimensional* node embeddings vectors $z_{G,i}^X \in R^d$ and $z_{G,j}^Y \in R^d$, respectively by using supervisory information of common friendship between u_i^X and u_j^Y in networks X and Y, respectively, along with structural information.

$$\begin{aligned} z_i^X &= f_{global}(u_i^X) \\ z_j^Y &= f_{global}(u_j^Y) \end{aligned} \tag{5.8}$$

If u_i^X and u_j^Y have more common friends, their embedding vectors $z_{G,i}^X$ and $z_{G,j}^Y$ are expected to be closer in embedding space. We employ a well-established network embedding DeepWalk [124] algorithm to preserve the local structure.

In order to obtain f_{global}, we need to view both graphs G_X and G_Y as single integrated global piece of information. More specifically, we create global graph G_{global} such that $V_{global} \subseteq V_X + V_Y$ and $E_{global} = CNL + NCNL$, where CNL represent the *cross-network linkages* referring to node pairs

which denotes identities in two social networks X and Y belonging to same individual and $NCNL$ represent the *non cross-network linkages* denoting identities known to be belonging to different individuals. We construct $NCNL$ as follows. For every node pair $(u_i^X, u_j^Y) \in CNL$, we perform a random walk of $t - depth$ starting at node u_i^X within social network X to get N_t^X nodes. Then for every $u_k^X \in N_t^X$, we add node pair (u_k^X, u_j^Y) to the set $NCNL$. Similarly, we construct node pairs in reverse manner starting with node u_j^Y in social network Y. This ensures that our G_{global} is closer to the real world scenario in which friends (nodes in 1-depth or 2-depth) of cross-network linkages are also considered. Besides, we also randomly sample node pairs r_i^X, r_j^Y such that they are not likely to have any common friends and add them to the set $NCNL$. Weights of all these edges (or node pairs) in both CNL and $NCNL$ are expressed in form of common friendship (CF) as depicted below.

$$w(u_i^X, u_j^Y) = CF(u_i^X, u_j^Y) \tag{5.9}$$

In order to learn node embeddings z_i^X and z_j^Y, we leverage the concept of performing *walks* in the *neighborhood*. For a given node $v \in V = V_X + V_Y$, we define the neighborhood as a set of nodes that are traversed in a walk starting from v, denoted by $N_S(v)$ using a walk strategy S. This walk strategy is guided by the *transition probability* (T_p) which defines probability to move to node v_2 starting at node n_1 and is computed as below.

$$T_p(v_1, v_2) = \alpha \times CF(v_1, v_2) \tag{5.10}$$

where α is the *search bias* which can be set to either 1 (no bias) or controlled using p and q (as done in *node2vec* [47]). Parameter p controls the degree of exploration while parameter q controls whether exploration happens in depth-first (DFS) manner or breadth-first (BFS) manner. The walk strategy is also dependent and biased by the *common friendship* (CF) between the nodes across which transition happens. Higher the common friends between two nodes across two social networks, more is the likelihood of walk traversing across them, and similar will be the node embeddings of such nodes in embedding space.

Optimization: In order to learn effective representative node embeddings, we make use of the concept of *skip-gram* architecture [110] which has been the foundations of prior state-of-the-art walk-based approaches namely *DeepWalk* [124] and *node2vec* [47]. Taking into account all nodes in the combined global graph $G = G_X + G_Y$, our objective is to maximize the log-probabilities of

finding node neighborhood of $N_S(v)$ for each node $v \in V_X + V_Y$ as below.

$$\begin{aligned} O_{global} &= \max_{f_{global}} \sum_{v \in V_X + V_Y} log(Pr(N_S(v)|f_{global}(v))) \\ &= \max_{z_i^X} \sum_{v \in V_X} log(Pr(N_S(v)|z_i^X)) \\ &+ \max_{z_j^Y} \sum_{v \in V_Y} log(Pr(N_S(v)|z_j^Y)) \end{aligned} \tag{5.11}$$

In order to make this objective function eq (5.11) computationally tractable we assume that the probability of hitting any nodes n_i in the neighborhood $N_S(v)$ is independent of hitting any other node in $N_S(v)$, which enables application of product rule as below.

$$Pr(N_S(v)|f_{global}(v)) = \prod_{n_i \in N_S(v)} Pr(n_i|f_{global}(v)) \tag{5.12}$$

And we model the probability of hitting node n_i, given the encoding vector representation of v by $f_{global}(v)$ as below.

$$Pr(n_i|f_{global}(v)) = \frac{exp(f_{global}(n_i) f_{global}(v))}{\sum_{u \in V_X + V_Y} exp(f_{global}(u) f_{global}(v))} \tag{5.13}$$

This optimization process coupled with the biased walks in proportion to the common friendship (CF) ensures that the resulting node embeddings are learned such that node pairs having higher CF values are closer to each other in embedding space.

5.4 Data Description

We evaluate our approach on two network datasets, one augmented, and another a real-world dataset to justify its applicability over a broad set of practical use cases. We extend a single network Facebook dataset and derive two subnetworks. We also acquire a popular real-world dataset that consists of users from two distinct social networks, Twitter and Foursquare. We adopt two different approaches to constructing the datasets. The first approach is based on generating two sub-networks from a given single-source network using the sampling method. The second approach is to construct datasets from two real-world social networks, namely Twitter and Instagram. While the first dataset generates *undirected* graphs, the second dataset comprises of *directed* graphs. We evaluate our proposed approach on both undirected and directed graphs to demonstrate its applicability in generalized settings.

5.4.1 Augmented Dataset

We use the Facebook friendship network dataset[2], provided by Viswanath et al. [154], comprising 63,713 users and 817,090 edges. We create an undirected graph from the dataset and filter out the nodes with a degree of less than 5, reducing the graph to 40,711 nodes and 766,579 edges.

Algorithm 1 Generate Sampled Dataset

procedure CREATE-SUBGRAPHS($G(V, E), \alpha_s, \alpha_c$)
    ▷ Take $G(V, E)$ as inputs and create two subgraphs $G_X(V_X, E_X)$ and $G_Y(V_Y, E_Y)$
    for $e \in E$ **do** ▷ Divide edges E into E_X and E_Y
        $p \leftarrow random(0, 1)$
        if $p < (1 - 2\alpha_s + \alpha_s\alpha_c)$ **then**
            discard e
        else if $(1 - 2\alpha_s + \alpha_s\alpha_c) < p < (1 - \alpha_s)$ **then**
            add e to E_X
        else if $(1 - \alpha_s) < p < (1 - \alpha_s\alpha_c)$ **then**
            add e to E_Y
        else
            add e to E_X and E_Y, both.
        end if
    end for
    return $G_X(V_X, E_X), G_Y(V_Y, E_Y)$
end procedure

We use this graph to create two subgraphs using a sampling algorithm proposed by Man et al. [102], depicted in 1. Given a graph $G(V, E)$, the algorithm takes two parameters, α_s, α_c and produces two subgraphs $G_X(V_X, E_X), G_Y(V_Y, E_Y)$. The parameter α_s (sparsity) represents how likely are the two subgraphs to retain the edges from the original graph, or the sparsity level. And, the parameter α_c (overlap) indicates the expected fraction of edges shared among the two subgraphs, or the overlap level. We run Algorithm 1 for the different values of α_s and α_c to get a variety of subgraph pairs suitable for our application that can emulate a real-world data. The four pairs of subgraphs are generated by α_s, α_c taking the values $[(0.5, 0.5), (0.5, 0.9), (0.9, 0.5), (0.9, 0.9)]$. Table 5.1 shows the number of edges and nodes in the generated subgraphs for different values of α_s and α_c. Once we have the subgraphs, we need to generate node pairs representing CNLs and NCNLSs across the two subgraphs, which we call as *X-node* pairs. To do so, we consider all the common nodes in both the graphs, $V_{CNL} = V_X \cap V_Y$, and call them as our CNL nodes, while we term others as NCNL nodes. Now, we take a CNL node and initiate a random walk of a variable length t in G_X, and later in G_Y. The random walks generate $2 \times t$ nodes from G_X and G_Y collectively, and these nodes are then paired with the CNL node to form node pairs. For each such pair, we calculate the *Common Friendship (CF)*, as shown in equation 5.7, and all the pairs that have $CF <= 0$ are

[2]http://socialnetworks.mpi-sws.org/data-wosn2009.html

Table 5.1: Statistics for the two datasets used for the evaluation.

Graph	#Nodes	#Edges	#CNLs
Augmented Dataset			
$G_X(\alpha_s = 0.5, \alpha_c = 0.5)$	40,558	383,463	39,061
$G_Y(\alpha_s = 0.5, \alpha_c = 0.5)$	40,563	382,380	
$G_X(\alpha_s = 0.5, \alpha_c = 0.9)$	40,562	383,360	40,458
$G_Y(\alpha_s = 0.5, \alpha_c = 0.9)$	40,547	383,528	
$G_X(\alpha_s = 0.9, \alpha_c = 0.5)$	40,602	422,295	40,418
$G_Y(\alpha_s = 0.9, \alpha_c = 0.5)$	40,708	689,481	
$G_X(\alpha_s = 0.9, \alpha_c = 0.9)$	40,709	689,856	40,705
$G_Y(\alpha_s = 0.9, \alpha_c = 0.9)$	40,709	690,103	
Real-World Dataset			
Twitter	5,120	130,575	1,288
Foursquare	5,313	54,233	

ignored.

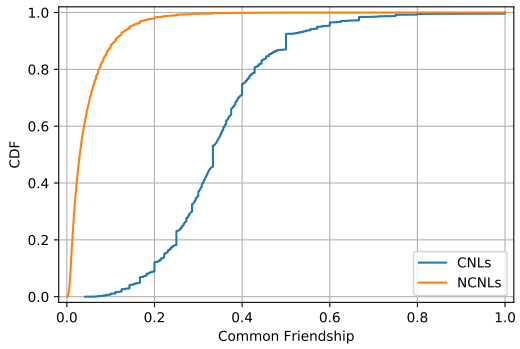
To study how the *CF* varies across with the chosen values of α_s (sparsity) and α_c (overlap), we plot a Cumulative Distribution Function (CDF) as shown in Figure 5.4 in different configurations. The *X-node pairs* with $\alpha_c = 0.9$ or higher overlap, tend to have more CNL pairs with higher CF values, as seen in Figure 5.4(b) and Figure 5.4(d). Similarly, keeping $\alpha_c = 0.5$, we see observe that *X-node pairs* with higher α_s have a relatively larger number CNL pairs with higher CF values. We don't notice a change in the distribution of CF values for NCNL pairs with the change in α_s and α_c values.

5.4.2 Real-World Dataset

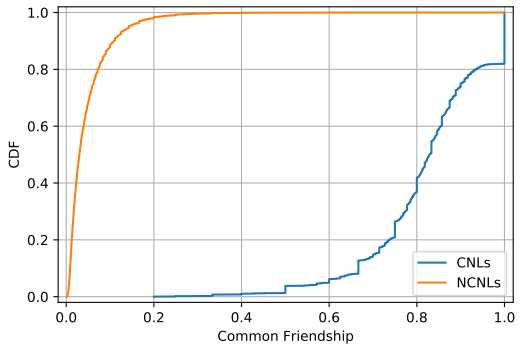
Kong et al. [78] introduced a network dataset collected from Twitter and Foursquare social networks. The data collection process is described in [78, 182] and used in multiple social link prediction problems [94, 184, 187]. Since the dataset comprises two graphs on its own, we do not need to employ any sampling algorithm to generate subgraphs, and we present the statistical details about the dataset in Table 5.1. The cross-network linkages represent the users that have profiles on both the social networks.

5.5 Experiments

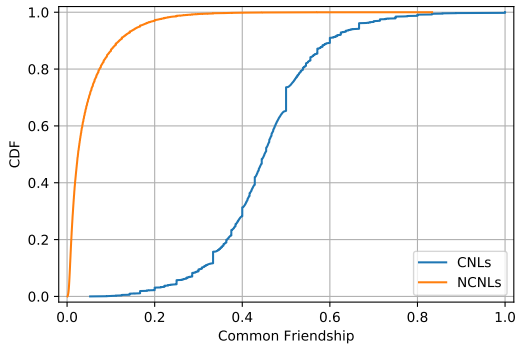
We design our experiments to answer the following research questions:



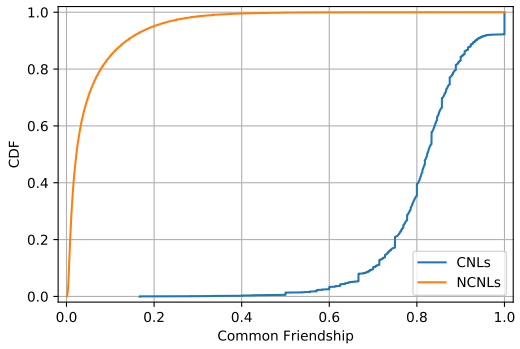
(a) Common Friendship CDF with $\alpha_s = 0.5$ and $\alpha_c = 0.5$



(b) Common Friendship CDF with $\alpha_s = 0.5$ and $\alpha_c = 0.9$



(c) Common Friendship CDF with $\alpha_s = 0.9$ and $\alpha_c = 0.5$



(d) Common Friendship CDF with $\alpha_s = 0.9$ and $\alpha_c = 0.9$

Figure 5.4: Common Friendship values for cross-network node pairs obtained from the augmented and the real-world dataset. Across all configurations, it is apparent that common friendship is less in Non-CNLs than in CNLs.

RQ1 How do the values α_s and α_c affect the retrieval of a cross-network node match?

RQ2 How does the choice of second node embedding function f_{global} affect the cross-network node retrieval?

RQ3 How does our proposed NeXLink framework compare with other baselines on a real-world dataset?

Experimental Setup

We implement all our experiments using NetworkX [51] for graph functions and use OpenNE[3] to run network embedding implementations. We perform experiments on a machine with CPU comprising of Intel(R) Xeon(R) Processor E3-1220v6 with 64GB RAM, and Nvidia GeForce GTX 1080 Ti GPU

[3]https://github.com/thunlp/OpenNE

with 12GB VRAM for CUDA-accelerated implementations for network embedding frameworks. To generate the $NCNL$ node pairs, we keep the depth of random walk, $t = 20$, throughout the experiments. When generating the embeddings for cross-network linkages, all embeddings functions treat node pairs as the edges of the cross-network graph, with CF values as the weights for cross-network edges. Given that our proposed NeXLink framework has two steps for the preservation of structure at the local and global level, we employ prior state-of-the-art node embedding methods at these steps. We typically employ LINE [148] to preserve the local structure and consider only first-order proximity calculated over first-order nodes and run over 50 epochs, with early stopping. We do not use second-order proximity since that is taken care of in the second step of our NeXLink framework. We employ various node embedding methods (LINE [148] and DeepWalk [124]) to preserve the global structure in the second step of our NeXLink framework. However, as we explain in this section, it turns out that node2vec [47] when employing common friendship across social networks, gives the best results. In node2vec, we set the parameters as $p = 1$ and $q = 2$, which, as mentioned by the authors, are more suited towards preserving structural equivalence. All embedding functions yield 128D embeddings.

Evaluation Metrics

We evaluate our approach to measure how effectively node embeddings preserve the CNLs in lower-dimensional space, and how closely network embeddings for CNL lie in that space. In order to compute closeness, we measure the cosine similarity over the node embeddings. In order to compute closeness, we measure the cosine similarity over the node embeddings. When querying for a node u_i^X from the CNL pair (u_i^X, u_j^Y), we count a hit if the matching node embedding z_j^Y for node u_j^Y is present in a set of k node embeddings, ordered on their similarity. To measure accuracy, we calculate a ratio of hits over number of queries and term it as *Hit-Rate@k*. *Hit-Rate@k* is defined as:

$$Hit(u_i^X) = \begin{cases} 1, & if\ z_j^Y \in \{z_1^Y, z_1^Y, ..., z_k^Y\} \\ 0, & otherwise \end{cases} \tag{5.14}$$

$$Hit - Rate@k = \frac{\sum_{i=0}^{N_{CNL}} Hit(u_i^X)}{N_{CNL}} \tag{5.15}$$

We choose $k = [1, 5, 10, 20, 50]$ for all the experiments to evaluate our approach under different budget values.

Baselines

We compare NeXLink against the most recent approaches that use structural information for network aligning using network embeddings.

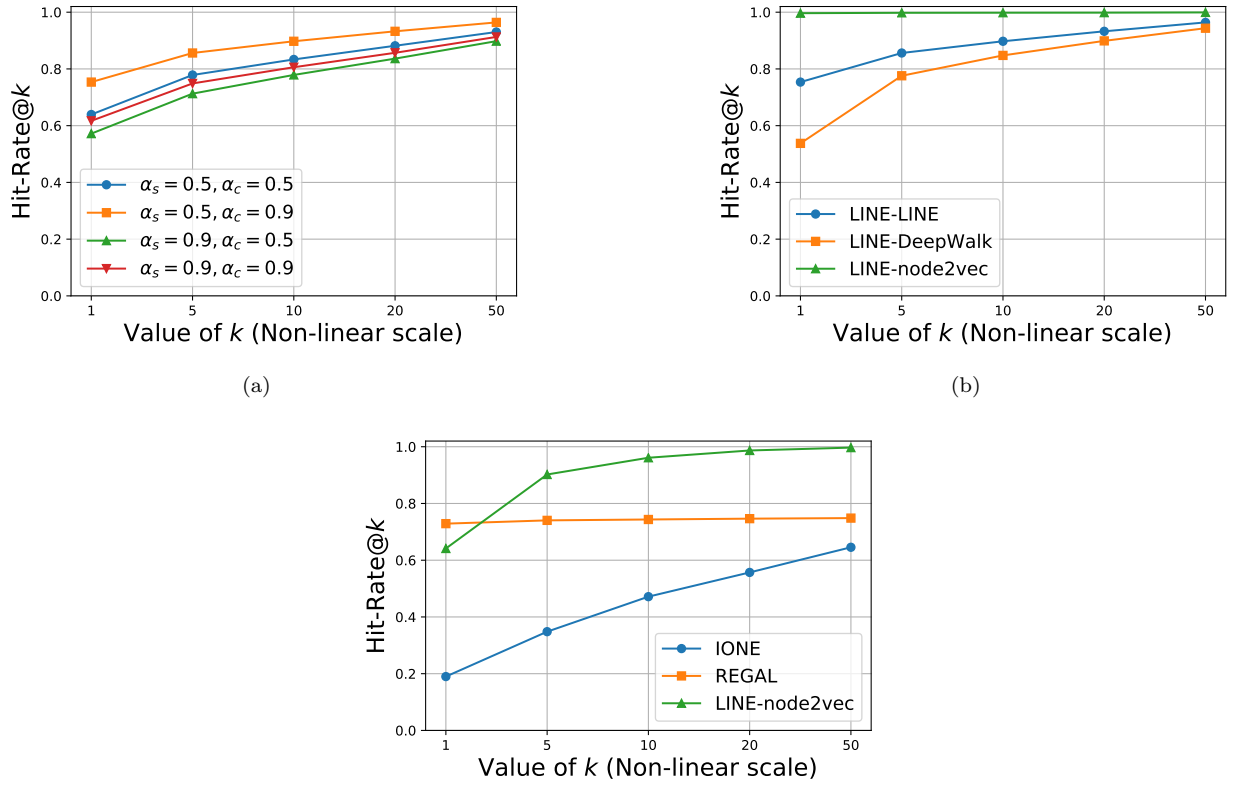


Figure 5.5: Results of the three experiments for our research questions (RQ1-RQ3). (a) Comparison of Hit-Rate@k values for different sparsity (α_s) and overlap (α_c) levels. (b) Comparison of Hit-Rate@k values for different cross-network node embeddings. (c) Comparison of Hit-Rate@k values for the baselines and NeXLink (LINE-node2vec) over the real-world dataset.

IONE [94] proposed Input-Output Network Embedding (IONE) for the task of network alignment by considering follower-followee relationships between the nodes and retaining those relationships by using *input* and *output* context vectors as node embeddings. They adopt stochastic gradient descent to learn the vector representations and compute the cosine similarity between them to map the users across the two graphs.

REGAL Representation Learning-based Graph Alignment (REGAL) proposed by [56] uses structure and attribute-related information to find similarities for network alignment. REGAL extracts node identities based on the degrees of its neighbors use singular value decomposition of a low-rank similarity matrix over landmark nodes to get node representation and greedily match these representations to find alignments.

5.5.1 Effect of Sparsity and Overlap levels

As seen in Figure 5.4, the α_s and α_c values affect the Common Friendship (CF) values for the CNL nodes, and since the second embedding function is trained to preserve the CF property across networks, we see significant differences in the performances with respect to the difference in α_s and α_c values. We start by employing LINE [148] to learn the local as well as cross-network similarity structure over the four subgraph configurations, as mentioned in Section 5.4, and present our results in Figure 5.5(a). We observe that the X-node pairs with $\alpha_s = 0.5$ and $\alpha_c = 0.9$ values achieve the highest Hit-Rate@k for all values of k, starting from 0.75 at $k = 1$, and up to 0.96 at $k = 50$. The X-node pairs with $\alpha_s = 0.9$ and $\alpha_c = 0.5$ values achieve the lowest Hit-Rate@k values with 0.57 at $k = 1$ and 0.89 at $k = 50$. We attribute this behavior to the fact that fewer edges and more overlap between the two subgraphs help the embeddings capture structural similarities with less noise. On the contrary, low overlapping edges and the total number of edges lead to a lower Hit-Rate@k value. We also observe that when α_s and α_c are set to equal values ($\alpha_s = 0.5, \alpha_c = 0.5$ and $\alpha_s = 0.9, \alpha_c = 0.5$), the *X-node pairs* don't have a discernible distinction in performance. Additionally, it is evident from the figure that Hit-Rate@k is directly proportional to the k value, as with increase in k, the probability of finding a cross-network node match increases.

5.5.2 Effect of Cross-Network Node Embedding

We continue with the highest performing *X-node pairs* from 5.5.1, and We study the role of different network embedding techniques in our proposed NetXLink framework help to preserve CNLs and their impact on the performance of the detection of CNLs across social networks. LINE [148] is suitable for a majority of the number of graphs that preserve local network structure through first-order proximity, which makes it an ideal choice of node embedding method for our within-network embeddings. Along with LINE, we use DeepWalk [124] over *X-node pairs* to get cross-network embeddings, as it uses the structural information about inter-connected nodes by performing truncated random walks to learn latent representations of nodes in a graph, which in our case would be CNLs across networks. Similarly, we employ node2vec [47], which proposes a flexible notion of node neighborhood by designing a biased random walk to learn feature representations of graph nodes. Figure 5.5(b) shows the results of our experiments with different node embeddings. The LINE-DeepWalk performs relatively low at $k = 1$, but reaches closer to the Hit-Rate@k of LINE-LINE at higher values of k. It can be explained as the DeepWalk algorithm uses a random walk to sample neighbors of a node to gather the structural information. However, it does not take into account the weights of the edges because it can not leverage the CF values for cross-network links. LINE-LINE performs relatively well as it preserves the first-order proximity proportional to the CF values and achieves a Hit-Rate@1 of about 0.75. However, using the bias parameters from node2vec to represent structural equivalence better, we gain a significant advantage over LINE-LINE and

LINE-DeepWalk to get a Hit-Rate@k of around 0.99 for most of the k values. By biasing the walk towards detecting cross-linkages and weighting the transition probabilities towards the CF values as per equation 5.10, LINE-node2vec gives an optimal representation of cross-linkages that are placed closer to each other in the embedding space.

5.5.3 Comparison with the Baselines

Finally, we evaluate our best performing combination of LINE-node2vec in the NeXLink framework with competing baselines. Along with the structural information, REGAL [56] allows using attribute information for node similarity. However, when comparing with our approach, we only use the structural information from the real-world dataset, described in 5.4.2. We also compare our approach with IONE [94] that takes two network graphs as input and produces node embeddings based on the follower and followee relationship among the nodes. We employ our best performing LINE-node2vec technique and elaborate on its performance on the real-world dataset. Figure 5.5(c) illustrates the performance of the baselines, as compared to our approach. Given the evaluation of IONE using the same dataset, we were to reproduce their results successfully, as mentioned in their work [94]. However, it still under performs when compared to the other approaches. REGAL achieves the highest Hit-Rate@1 as it uses node degrees to capture structural similarities, and node degrees partially contribute to the CF values. However, it still fails to completely leverage the essential CF values, as one of its limitations is not being able to take the edge weights into account. Therefore, its performance stagnates at higher k values. In contrast, LINE-node2vec starts below REGAL at $k = 1$, but achieves higher Hit-Rate@k values with the increase in k. LINE-node2vec learns both within-graph and cross-graph structural features from the real-world dataset and effectively represents the similarities in low-embedding space.

Given that our proposed NeXLink framework makes use of LINE [148] and node2vec [47] algorithm, so the time complexity is given by $\mathcal{O}(dk|E| + r|V|l)$. Table 5.2 compares time complexity with the

Table 5.2: Comparison of algorithmic complexity of LINE, REGAL, and NeXLink.

Algorithm Name	**Complexity**
LINE [148]	$\mathcal{O}(dk\|E\|)$
REGAL [56]	$\mathcal{O}(\|V\|max\{kd^2, pb, p^2, log(n)\})$
node2vec [47]	$\mathcal{O}(r\|V\|l)$
NeXLink	$\mathcal{O}(dk\|E\| + r\|V\|l)$

other baselines. The first component $\mathcal{O}(dk|E|)$ comes from LINE where d represents number of dimensions in the embedding vector, k represents number of negative samples, and $|E|$ is the number of edges in the graph. The second component comes from node2vec where r represents number of iterations, l represents the length of random walks, and $|V|$ is the number of vertices in the graph.

The space complexity of our proposed NeXLink framework is $\mathcal{O}(d|V|)$, because each vertex in the set of vertices $|V|$ is represented by d-dimensional embedding vector.

5.6 Limitations and Discussions

While developing NeXLink, we identify some of the limitations of our approach. Firstly, we only include structural information indicating standard connections in the two networks, to learn node representations. We can utilize more rich features to gain more comprehensive node representations. Secondly, an essential step in our approach is to create cross-network pairs, which we accomplish using random walks. We can evaluate more efficient ways to sample the cross-network pairs. And last, the two significant limitations of node embeddings are (a) the need to define an objective function, based on which we learn the embeddings, and (b) node embedding models are transductive, which means that it is not possible to generate the embeddings for the nodes that we do not see during the training. To this end, we can consider the use of graph neural networks [53, 135].

In this chapter, we propose our *NeXLink* framework for the effective representation of cross-network linkages across social networks. Our framework works by preserving the local structure of nodes within the same social network and global structure manifested in the form of common friends exhibited by nodes participating in cross-network linkages. We perform an extensive evaluation of our approach on two datasets, one of which we augment from the Facebook social network, and the other comprises of Twitter-Foursquare node pairs. Given that the NeXLink framework is flexible, we explored numerous state-of-the-art node embedding algorithms and found that LINE-node2vec performs the best when provided with supervisory information of common friendship. It performs with an average Hit@1 rate of 98% across all configurations of the augmented dataset. Further, our approach outperforms state-of-the-art node representation algorithms LINE and DeepWalk for representing cross-network linkages across the social networks. This can be attributed primarily to the fact that our approach preserves local and global cross-network links more effectively than these previous approaches, specifically targeted to perform well on single networks. Our framework works better than other state-of-the-art node embedding approaches like IONE and REGAL for identity linkage on a real-world dataset. This is because our framework performs biased walks in accordance with the common friendship metric for cross-network links. As future work, we can include node attributes derived from user profile configuration and user content in the NeXLink framework and their impact on performance measured. Deep learning-based approaches for node embedding would also be the right direction to explore at the algorithmic level.

Chapter 6

Nudging Nemo: Helping Users Control Linkability

We have discussed different data collection approaches to collect linked user identities, described about biases in the datasets, and subsequently proposed a method to link user identities across social networks. However, the ability to link different identities, referred to as *linkability*, poses a threat to the users' privacy; users may or may not want their identities to be *linkable* across networks. Therefore, in this chapter[1], we propose *Nudging Nemo*, a framework that assists users in controlling the linkability of their identities across multiple platforms. We model the notion of linkability as the probability of an adversary (who is part of the user's network) to link two profiles across different platforms, to the same real user. Nudging Nemo has two components; a *linkability calculator*, which uses user identity linkage methods to compute a normalized linkability measure for each pair of social network platforms used by a user, and a *soft paternalistic nudge*, which alerts the user if any of their activity violates their preferred *linkability*. We evaluate the nudge's effectiveness by conducting a controlled user study on privacy-conscious users who maintain their accounts on Facebook, Twitter, and Instagram. Outcomes of user study confirmed that the proposed framework helped most participants make informed decisions, thereby preventing inadvertent exposure of their personal information across social network services.

[1]Mostly taken from our published paper. **Rishabh Kaushal**, Srishti Chandok, Paridhi Jain, Prateek Dewan, Nalin Gupta, Ponnurangam Kumaraguru. Nudging nemo: Helping users control linkability across social networks. In *Proceedings of International Conference on Social Informatics, 2017*.

6.1 Introduction

Users join multiple Online Social Media [2] (OSM) platforms because they offer different types of content and network (friends) to users. Some OSMs promote sharing of images (like Flickr and Instagram) or videos (like YouTube) while others promote sharing of short messages (like Twitter) or a combination of messages, video, and images (like Facebook). Some OSMs provide access to the professional network (like LinkedIn) while others provide access to a more personal network (like Facebook). These factors complicate and affect users' participation in these networks. For instance, an incoming friend request on a professional network tends to be accepted even if a requester is not personally known (referred to as 'others') whereas, on a personal network, a user would not like to accept such a request. Similarly, a user is likely to post about personal life events on a network like Facebook, but would probably refrain from doing the same on a professional network like LinkedIn [142, 153].

Most instances discussed above are commonplace for a majority of social media users today. However, such instances give rise to a variety of privacy implications which are seldom addressed or acknowledged. Consciously or unconsciously, users tend to have a certain set of attributes and characteristics common across multiple social media platforms (for example, date of birth, city of residence, screen name, etc.), which enables linkage of two profiles on different platforms belonging to the same real-world user. We have termed this concept of linking two online profiles to a user as *identity resolution* in this Chapter (which means the same as *user identity linkage*), and have demonstrated multiple techniques in the past where they have been able to correctly link profiles across platforms with a high success rate [8, 19, 41, 65, 67, 87, 90, 93, 95, 175], which we have discussed in Chapter 2 of this thesis.

In this work, we propose *Nudging Nemo*, a framework that allows users to learn about and control the linkability of their profiles across different social media platforms. Our key contributions are as follows:

- We quantify linkability using a metric termed as *linkability score*, which quantifies either separation or closeness between two identities belonging to the same user on different OSM platforms. Such a metric empowers the user to control his linkability across OSM platforms.
- We identify the factors (profile attributes) that contribute to the computed linkability score so that the user is well informed about taking remedial measures.
- We design and develop a soft paternalistic *linkability nudge*, which alert users whenever their behavior results in a change of linkability score beyond the user-configured desired range.

[2] We use the term OSN and OSM, interchangeably in this thesis. More specifically, OSN is referred to platforms which emphasize on networking among users, and OSM platforms focus on the content.

- Lastly, we conduct a controlled lab study to evaluate the effectiveness of the linkability nudge.

The rest of the chapter is organized as follows. Section 6.2 describes the preliminaries, which includes attack scenario, scope, assumptions, and approach to the solution. Work related to methods for Identity Resolution (IR) and privacy nudges is described in Section 6.3. Subsequently, in Section 6.4, we design and develop our proposed system for computing linkability scores by leveraging well-known IR methods. In Section 6.5, we explain in detail architecture, design, and features of proposed linkability nudge. This is followed by the user evaluation of nudge and results in Section 6.6. Finally, in Section 6.7, we discuss the implications of our work, conclude our work, and outline future work.

6.2 Preliminaries

Users are creating multiple identities across OSM platforms for various reasons outlined earlier. Depending upon their requirements and needs, users would like to reduce the linkability of their multiple identities to prevent unintended exposure of personal behavior on one OSM. In this section, we discuss the attack scenario and our assumptions.

6.2.1 Attack Scenario

We presume that adversary (an entity who wants to link two profiles) would have access to victim's (user who's profiles are under consideration) identity on at least one OSM platform (say i_A) and would subsequently use one or more of the multiple variations of identity resolution as below.

1. Given a pair of identities i_A and i_B on two OSM platforms sites A and B, respectively, the goal is to find a function that returns 1 or 0 depending upon whether i_A and i_B belong to the same user or not, respectively.

2. Given a single identity i_A on OSM platform A and candidate set of identities C_B on OSM platform B for the same user, the goal is to find a function which identifies correctly i_B from within C_B (searching problem).

3. Given a matching set of identities C_A and C_B on two OSM platforms A and B, respectively, the goal is to find all pairs of identities (i_A , i_B) which belong to the same user.

The adversary would typically implement well-known methods that solve identity resolution problem taking i_A as input and obtain i_B of the victim in the OSM platform B in which the adversary is not connected to the victim. This implies that an adversary could be a friend of the victim in a

professional network and use identity resolution methods to identify victims in a personal network, thereby gaining access to victim's activities in the personal network.

6.2.2 Assumptions & Scope

Our work takes into account the following assumptions and scope.

1. There are privacy-conscious users [81] who maintain multiple identities across OSM platforms and desire to keep their identities on at least one pair of OSM platforms as far as possible, in other words, does not want their identities to be resolved through identity resolution attack described above.

2. Such users would be interested to know how *linkable* their identities are. This can be expressed in a quantifiable metric for each pair of identities on OSM platforms.

3. The adversary is intelligent enough to implement automated methods for identity resolution, thereby capable of resolving identities at a large scale over a period of time. On the other hand, privacy-conscious users (victims) may not have the capability to protect themselves against automated identity resolution attacks.

6.3 Related Work

Numerous methods and techniques have been studied by researchers for performing identity resolution (also referred to as user identity linkage) across multiple OSNs, which we have discussed in Chapter 2 of this thesis. Besides these, we also draw ideas from prior work related to nudge's design, particularly those related to privacy. Leenes et al. [84] in their work suggested segregation of the audience for profile attributes of users on OSNs so that its visibility is controllable. Wang et al. [160] designed and implemented modifications to the Facebook web interface that would nudge users to consider the content and audience of their online disclosures. Wang et al. [162] had also earlier developed three types of privacy nudge, one was to provide the audience of a post, second was developed to introduce time delays before a post goes public and third was provided to obtain user feedback. Authors in [161] and [163] worked to understand and find out the set of actions that users perform over OSNs, which they later regret which could be a good indicator of privacy leaks and need for nudging so that those actions do not get repeated in future. Ziegeldorf et al. [196] proposed a novel design paradigm called *comparison based privacy* in which users can compare their privacy metrics with other groups of users to evaluate privacy disclosure levels. From works of [4] and [178], it can be seen that with the widespread use of mobile devices, the ideas of privacy nudges are being applied on mobile platforms as well. To the best of our knowledge, there is no

prior work that provides a mechanism of nudging (or providing feedback) users to help them control the linkability of their identities, which is the focus of our work in this chapter.

6.4 Linkability Score

The linkability score quantifies the degree of closeness or separation between two identities on a pair of OSM platforms. Linkability score varies between 0 to 1, and a lower value would mean that the two identities are less linkable where high value would mean more linkable. Our approach to solution comprises computing a function that takes a user's identities i_A and i_B on two OSM platforms A and B, respectively, as input and compute linkability score between them as below.

$$LS_{i_A,i_B} \leftarrow f_{linkability\_score}(i_A, i_B) \tag{6.1}$$

Identity of a user u on OSM platform X is modeled as feature vector that is values $< v_1^X, v_2^X, ..., v_n^X >$ corresponding to n features $< f_1^X, f_2^X, ..., f_n^X >$. Given an identity pair $< i_A, i_B >$ as input, the function for computing the linkability score is weighted sum of appropriate *feature similarity metric* (FSM) between corresponding feature values of identity pair.

$$f_{linkability\_score}(i_A, i_B) \leftarrow \frac{1}{n} \sum_{i=1}^{n} FSM(v_i^A, v_i^B) \tag{6.2}$$

We adopt a uniform weight sum formulation, thereby giving equal importance to all features towards the linkability score, and leave improved formulations for future work. However, we do *rank* features based on their contribution to the linkability score. Both, the linkability score and ranked feature information would be useful to a privacy-conscious user who would like to keep the linkability score to a lower value. Subsequently, a system referred to as *linkability nudge* is designed, developed, and evaluated, which makes soft paternalistic interventions (nudges) whenever a user behavior causes a linkability score to go beyond the desired range of linkability score.

6.4.1 Design & Implementation

In order to compute linkability scores between pair of identities of a user, we design a web based application based on Django framework.[3] Fig 6.1 depicts the flowchart of the steps which are performed for computation of linkability scores.

On *client side*, there are two key steps as below.

[3]Django Framework, https://www.djangoproject.com/

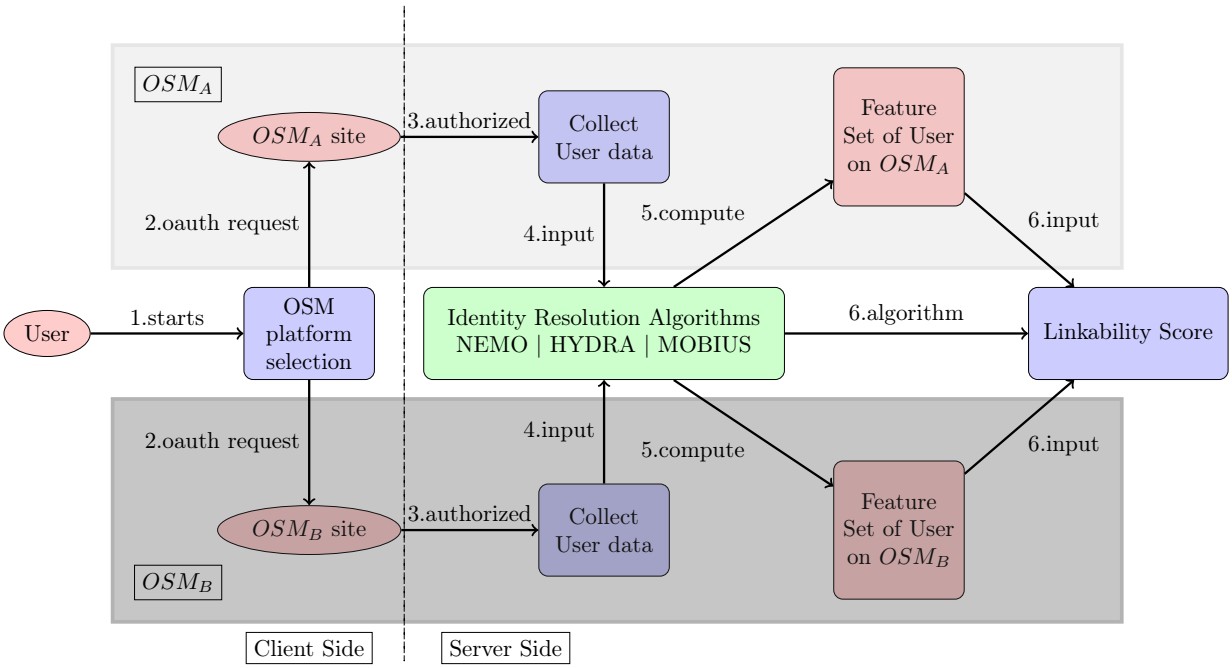


Figure 6.1: Flowchart depicting the steps involved for computing linkability scores.

1. User selects the OSM platform, in our experiments, as we shall discuss later, the options are Facebook, Twitter, and Instagram.

2. User sends a request for grant of access token so that our web application can acces user's profile information.

On *server side*, following steps are performed.

1. After obtaining access authorization, the web application collects user's data from the OSM platform's API endpoints.

2. Collected user data is passed as input to identity resolution algorithms, which specifies various features, say $< f_1^X, f_2^X, ..., f_n^X >$.

3. Using the user's data, values of these features are computed on different OSM platforms, say $< v_1^X, v_2^X, ..., v_n^X >$.

4. Finally, using the feature vectors and algorithm (namely, Nemo, Hydra, and Mobius), linkability scores for each pair of OSM platforms are computed using equation 6.1 and eq. 6.2.

6.4.2 Identity Resolution Methods

We leverage features from three well known Identity Resolution (IR) methods namely Nemo [67], Hydra [95] and Mobius [175]. All these methods propose techniques using the user's profile at-

tributes and behavior to resolve user's identities across OSM platforms. However, we aim to build upon these existing IR methods and propose a metric which we refer to as *linkability score*, which quantifies the possibility of linkability or non-linkability of user's identities across OSM platforms. In the first IR method used, referred to as **NEMO**, Jain et al. [67] have used four algorithms for identity resolution, namely profile search, content search, self-mention search, and network search. In our work, we have used only *profile search* and *content search* algorithms. For computing linkability between two identities on different OSMs, we have considered five features, namely username, name of the user, location, profile image, and post contents with suitable similarity measures. In the second IR method that we use, referred to as **HYDRA**, Liu et al. [95] have mainly considered user behavioral modeling, namely, User Attribute Modeling, User Style Modeling, and Multimedia Content Generation. User Attribute Modeling considers textual attributes and visual attributes configured in their identities by users on different OSM platforms. To sum up, we use the name of the user, education, work, profile image, website, post contents, and multimedia content (images) to as the features. The third IR method, referred as **MOBIUS** and proposed by Zafarani et al. [175], is based on the fact that when individuals select usernames, they exhibit certain behavioral patterns, which often leads to information redundancy. We computed the top 10 most important features identified by Zafarani et al. for username matching in the context of identity resolution. It may be stated here that due to restrictions in the endpoints offered by APIs and the number of attributes offered by OSM platforms, namely Facebook, Twitter, and Instagram, we could use only limited features of NEMO, HYDRA, and MOBIUS.

6.4.3 Ethics

Given that we are accessing user's data, we have taken the utmost care that we follow the principles of ethical research. The user data we collect is obtained using temporary access tokens, which would typically expire after a few hours, and we would no longer be able to get user data anytime in the future unless the user explicitly refreshes them. All users involved in the evaluation of our nudge were informed about data collection and data usage upfront; they were recruited in the evaluation study voluntarily.

6.5 Linkability Nudge

Linkability nudge is our proposed mechanism that introduces soft paternalistic interventions to the user whenever the user's behavior causes the linkability score to change beyond the desired range configured by the user.

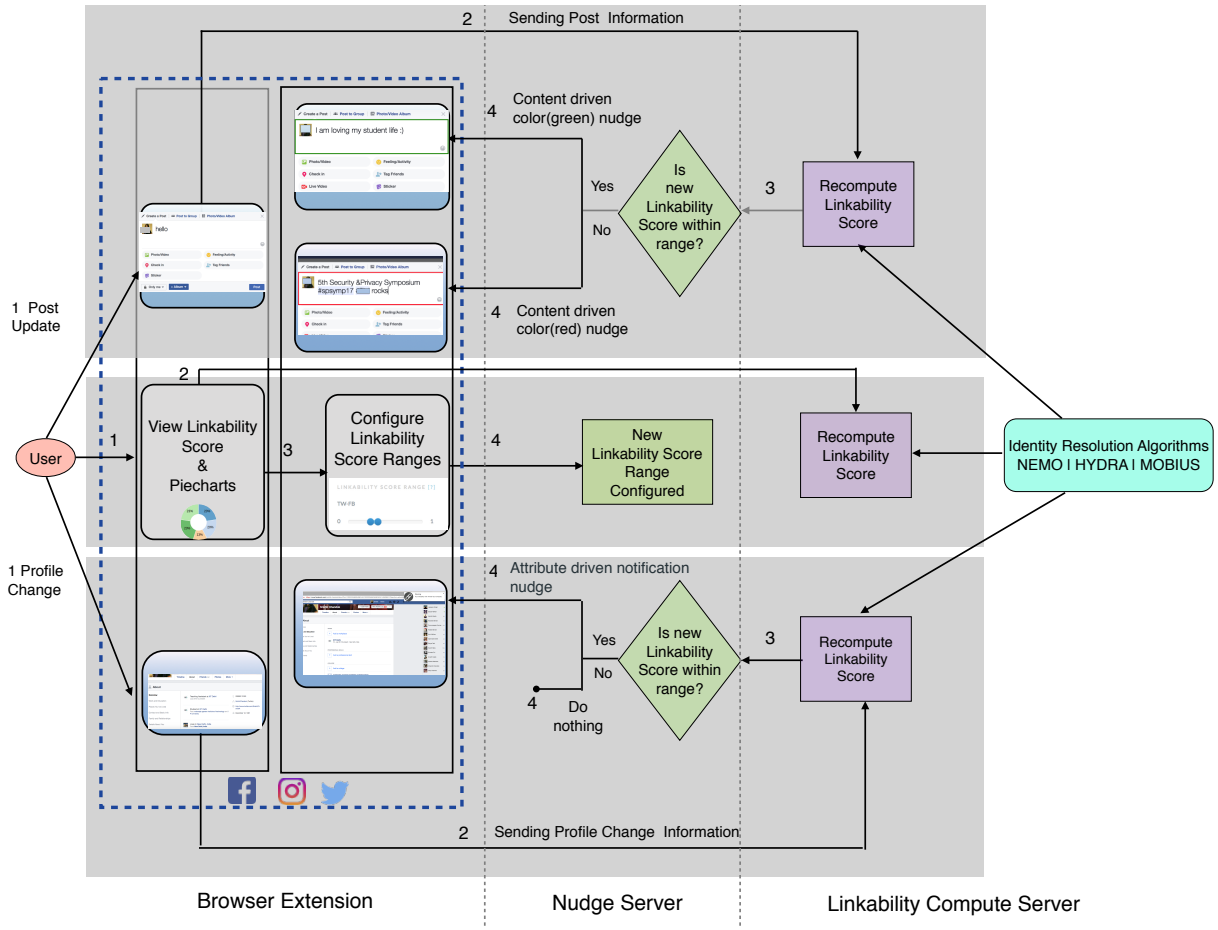


Figure 6.2: Flow diagram of operation of Linkability Nudge depicting three key components namely browser extension(plugin), Nudge Server and Linkability Compute Server. After post update, the screenshots are depicted later in Figure 6.3 as Content-driven Color Nudge. After profile change, the screenshots are depicted later in Figure 6.4 as Attribute-driven Notification Nudge.

6.5.1 Architecture

We implement linkability nudge by developing a chrome browser *plugin* that can be installed on user's web browser.[4] This plugin monitors user's behavior in terms of the content being posted over OSM platforms and changes to profile attributes being made on OSM platforms. Architecturally, linkability nudge comprises three main components, namely browser extension, nudge server, and linkability compute server, as depicted in Fig. 6.2.

Browser Extension This is the only component where a user is required to install on Google chrome web browser. It performs several functions as follows: (1) Maintains the user's identity

[4]Plugin shall be soon made available on Chrome Web Store for people to use and provide their feedback.

and user context across the entire user session. (2) Captures user's posting activity and changes in profile attributes on all configured OSM platforms. (3) It also displays linkability nudge in various forms, discussed later.

Nudge Server: This is the component that is required to be installed on the server side. It is an intermediary which sits between the *browser extension* and *linkability compute server*. It performs the following functions: (1) Receives user's access token from browser extension and sends them to OSM servers to obtain user's data. (2) Stores user's data in a database temporarily. (3) Passes the information pertaining to the user's activities like making a post or changing profile attribute to the linkability compute server. (4) Sends across the newly computed linkability scores to the browser extension from time to time based upon user's activities.

Linkability Compute Server: This is the component that performs most of the heavy computation involved in the calculation of linkability scores, and it is to be installed on the server side. It performs the following functions: (1) It implements the identity resolution methods to compute linkability scores. (2) It retrieves the user's data from the database as input to compute linkability scores at initial setup time. (3) Subsequently, it receives every user activity's information (whether making a post or changing profile attribute), recomputes linkability scores, and sends them back to the nudge server.

6.5.2 Nudge Design

Inspired from the works of Schaub et al. [136] and Acquisti et al. [1] for designing privacy notices and nudges, in our proposed nudge design, we have focused on two types of nudges.

Content-driven Color Nudge: Users having identities across multiple OSM platforms often indulge in *cross posting*, which means posting the same or similar information across multiple OSM platforms. Such behavior increases similarity in their identities, thereby increasing the linkability score. Our first nudge design addresses this particular issue by nudging the user through the use of color. Whenever a user types a post similar to any of the existing posts made by the user on other OSM platforms, we nudge the user by coloring the post's text box border with red shown in Fig 6.3(a). Color is green as long as linkability scores are within their pre-configured ranges, as shown in Fig 6.3(b). This is an indication to the user that this post is an instance of *cross posting*, which is going to increase the user's linkability across OSM platforms. Nudge is only a soft paternalistic intervention. We leave the text box colored with red and let the user decide whether he/she wants to continue making the post or refrain from making the post.

Attribute-driven Notification Nudge: User with multiple identities across OSM platforms maintain their identities such that there is overlap among the values of attributes specified by them on these OSM platforms. More the overlap, the more similar the identities would be, and higher

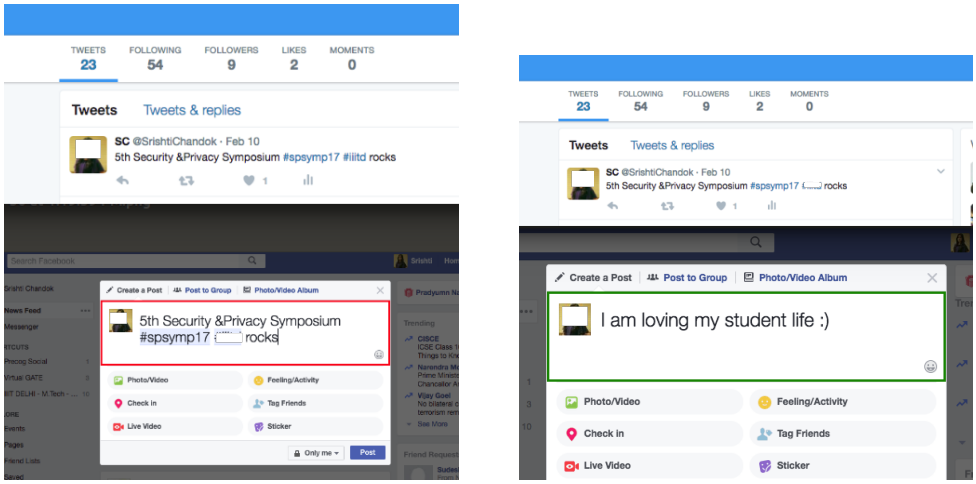
would be the linkability scores. In fact, the initial linkability scores are computed when the user grants authorization is mostly due to similar profile attributes like name, username, location, and profile picture. Whenever a user modifies the value of any profile attribute over an OSM platform, which causes a change in linkability score such that the score goes beyond the pre-configured desired range, then the user is nudged. Nudge is delivered in the form of a pop-up notification on the top right of the screen with a short message saying *'Your linkability with Facebook has increased'* as shown in Fig 6.4. Again here, being only a soft paternalistic intervention, we allow users' change in attribute to take place and let users decide whether the user wants to revert the change or not.

6.6 User Evaluation & Results

In this section, we present our approach for evaluating the system of linkability nudge by performing a controlled lab study.

6.6.1 Participants

To gauge user's perceptions and opinions concerning usage and linkability issues in a multi-OSM scenario, we engaged 40 participants in *pre-study questionnaire*. Subsequently, we filtered out and recruited only 12 participants for controlled lab study who had their accounts on all the three OSM platforms (namely Facebook, Twitter, and Instagram) on which our proposed linkability nudge was



(a) Facebook post is similar to Twitter post, the text box around post shows up in red.

(b) Facebook post is different from Twitter post, the text box around post shows up in green.

Figure 6.3: Illustration of Content-driven Color Nudge in which it is assumed that user has already made a post on Twitter and then is making a post on Facebook.

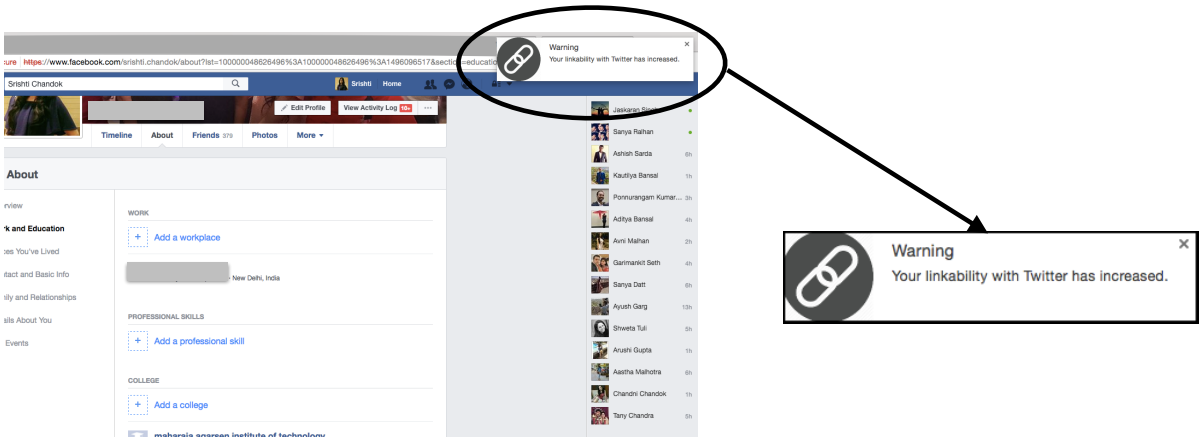


Figure 6.4: Illustration of Attribute-driven Notification Nudge on top right of the Facebook page alerting user with a short message that 'Your linkability with Twitter has increased', similar notifications are present to user on interfaces of Twitter and Instagram. Also shown is the enlarged view of nudge notification.

designed. Participants were within the age group of 18-26 years, with 67% female and 33% male comprising of mostly undergraduate students studying computer science.

6.6.2 Study Design

We conducted controlled lab study in two phases namely

- *Control Period*: Participants are not exposed to linkability nudge. They are asked to perform tasks, as outlined in the next section.
- *Treatment Period*: Participants are subjected to linkability nudge. In this phase again, we ask the participants to perform the same tasks as performed in the control period.

6.6.3 Tasks

In order to prompt the user to perform some activities so that effect of linkability nudge could be observed, we designed two types of tasks: (a) Making *scenario based posts* in which users are asked to make a post for a given hypothetical scenario and (b) Changing *profile attributes* for the identities maintained by users on OSM platforms. Detailed task descriptions are not mentioned in this paper owing to space constraints.

6.6.4 Results

Here we present our observations and outcomes of user interactions with linkability nudge, the nudging patterns on the users, user behavior, and overall user evaluation. To help us in all of these, we plotted activities of all participants on a timeline from start of experiment till end including both control and treatment period, total of around one hour as depicted in Fig 6.5.

Interactions with Nudge

The timeline plot helped us understand the user's interactions with linkability nudge (degree of participation) and vice-versa (nudging frequency).
Degree of Participation: Based on the amount of time spent and the number of tasks performed (shown in Fig 6.5) both during the control and treatment period, we can divide participants among three categories. P1, P3, and P6 performed at least eight or more tasks, taking into account both scenario-based posts (shown in + symbol) and profile changes (shown in $\times$ symbol) during treatment period, we consider them *highly active*. While P4 and P5 also spent the entire duration of one hour, but they performed very less number of tasks during the treatment period. P10 and P12 performed reconfigurations in their linkability scores (shown in $\star$ symbol) and were *moderately active*. While the remaining participants performed at least two tasks and were *least active*. We recorded passive activities of participants in which they viewed their linkability scores (shown in $\triangleright$ symbol) and factors contributing to those scores in the form of piechart (shown in $\circ$ symbol).
Nudging Frequency: Participants were nudged during the treatment period while during the control period, they were not nudged (in Fig 6.5, the transition from control to treatment period is depicted by a | symbol). Content-driven color nudge is depicted by either $\bigtriangledown$ (red) symbol or $\bigtriangleup$ (green) symbol while Attribute-driven notification nudge is depicted by $\square$ symbol. Participants who were *highly active* were also nudged the most, more specifically P1, P3, and P6 received nudges 10, 13, and 7 times, respectively. Participants who were *moderately active* received at least twice while the *least active* ones were nudged at least once.

Impact of Nudge on User Behavior

We may recall that nudge is an intervention that makes users more informed so that they may make better decisions. By design, nudges are suggestive and not binding on a user. Consequently, we observed that at times users did change their behavior while at other times, they overlooked the nudge.

Impact of Content-driven Color Nudge: From Fig 6.5, we see that both participants P11 and P12 in their last activities tried to make a post after which they were prompted with a content-driven red color nudge (+ symbol followed by $\bigtriangledown$ (red) symbol), and they refrained from making the post.

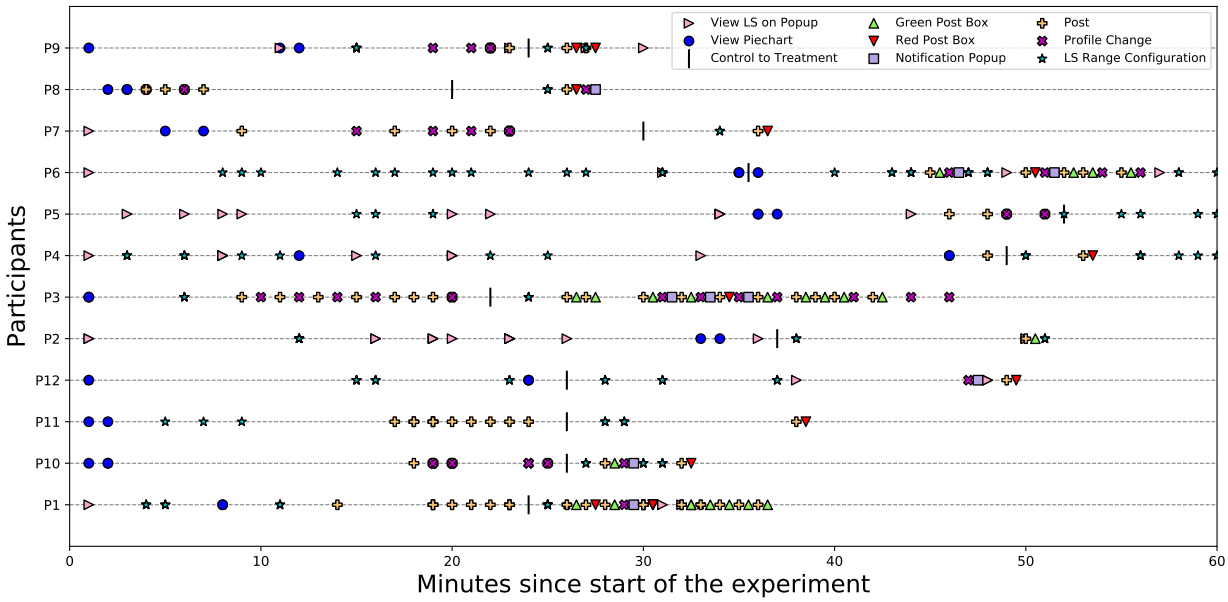


Figure 6.5: Complete timeline of activities of all 12 participants who took part in controlled lab study performing various tasks in control and treatment period. LS in the legend stands for linkability score.

In contrast, participant P1 continued to make a post even when content driven red color nudge was displayed (+ symbol followed by $\bigtriangledown$ (red) symbol, which is again followed by + symbol indicating that participant continued to make the post).

Impact of Attribute-driven Notification Nudge: From Fig 6.5, we see that participants P6 and P10 performed a profile change which triggered notification nudge which is immediately followed up by them to make a change in linkability score range ($\times$ symbol followed by $\square$ symbol followed by $\star$ symbol). In contrast, participant P3 made a number of profile changes and was shown notification nudge, which was ignored (in other words, linkability score was not reconfigured; neither was profile change undone). P12, after having shown notification nudge, only viewed linkability scores.

Implications of Nudge

To understand the overall impact of the nudge, we assess its efficacy on two parameters namely in creating awareness and usefulness.

Awareness of Linkability: 58% of participants (7 out of 12) understood the concept of linkability either completely or most of it after using our proposed linkability nudge while the remaining 42% said that they understood a little bit about it. 42% of participants (5 out of 12) said that they are absolutely sure that they are more aware of linkability implications and better informed after using the nudge while another set of 5 respondents said that they are 'somewhat' more informed. Most of the participants (84%, 10 out of 12) said that they did notice the factors contributing to

their linkability scores which itself suggest that participants were well informed about the causes for their linkability scores.

Nudge Utility: With respect to the utility of nudge, we found that the most popular among users was the *Content-driven Color Nudge*, which was liked by almost 84% of the participants (10 out of 12). This was followed by piecharts showing the contribution of profile attributes towards the linkability score, which was liked by 75% of participants (9 out of 12). In terms of the overall assessment of participants with respect to *usability* of the proposed linkability nudge, 58% (7 out of 12) found it to be useful and easy to use, while 33% (4 out of 12) found it useful but complicated for use and only one participant did not find it useful.

6.7 Discussion and Limitation

The purpose of linkability nudge was to help users understand the nuances involved in the linkability of their identities across OSM platforms. The goal is that when they perform an activity (making a post or changing profile attribute), they are conscious of the fact that it may increase or decrease the linkability of their identity concerning their identities on other OSM platforms. Linkability nudge would be most beneficial to those who often make personal posts on one network and do not want their colleagues to identity them on personal networks. Preventing linkability at the level of *profile* is quite challenging, given that users would prefer to have similar values in their profile settings. However, our proposed linkability nudge goes beyond and takes into account linkability at the level of *content* being posted as well. Participants of user study exhibited a varied level of participation and were intervened by all types of nudge designs during the controlled lab study. It is evidently clear that the behavior of at least some of the participants did change when they were exposed to linkability nudge. They either refrained from making a post, which is increasing their linkability or reconfigured the linkability score ranges. On the other hand, some of the participants' behavior did not change, which suggests that they were not concerned about linkability issues. We expected more activities from the participants, and in the future, we would explore ways to improve it. Most of the participants liked the color nudge reinforcing the notion that simple designs make a significant impact.

Linkability nudge was able to make most of the participants more aware of the linkability issues. Some participants expressed concern over complicated usability; on further investigation, we found that it was mainly due to the time delay (2-5 seconds) they experienced while making post during the treatment period. This is because each word typed is sent back to the server for re-computation of the linkability score, causing the delay. We shall work to improve the engineering design to reduce the delay. We plan to deploy our proposed system of linkability nudge in the public domain and

conduct a field study to understand its impact more extensively on a wider audience. To conclude, we may say that users maintaining multiple identities across OSM platforms can see, in quantifying terms, the linkability of their identities between each pair of OSM platform. Linkability nudge helped users to take corrective measures to avoid inadvertent disclosure of their personal information owing to increased linkability. User evaluation validates that linkability nudge is indeed quite helpful in making users understand the concept of linkability and helps them through soft interventions to remain within their desired linkability ranges.

Chapter 7

Application: Clone Detection using User Identity Linkage

In this chapter[1], we present an application of the problem of user identity linkage. Rather than finding user identities belonging to the same user across the two social networks, we focus on finding *similar-looking* user identities within the same social network, which we refer to as *clone identities*. By similar-looking, we mean identities which impersonates the victim's identity, in a typical impersonation attack [143]. The account registration steps in Online Social Networks (OSNs) are simple to facilitate users to join the OSN sites. Alongside, Personally Identifiable Information (PII) of users is readily available online [144]. Therefore, it becomes trivial for a malicious user (attacker) to create an impersonated identity of a real user (victim), referred to as clone identity. While a victim can be an ordinary or a famous person, we focus our attention on clone identities of famous persons (celebrity clones). These clone identities ride on the credibility and popularity of celebrities to gain engagement, impact, and at times indulge in malicious activities. To address the issue, in the first part of this chapter, we build an automated clone detection model to identify the clones of a given victim. This approach is quite similar to solution approaches for user identity linkage, the difference being that we are detecting identities exhibiting similar features within the same social network. We evaluate our clone identity detection approach on 1,614 identities using 13 features and achieve 86% accuracy with precision and recall of 88% & 83%, respectively. In the second part of the chapter, we build a model that automatically characterizes the behavior of clone identities into five categories based on the content being posted by them: promotion, indecency, attention-seeking, advisory, and opinionated of which are benign while others are malicious. To this end, we extract an exhaustive set of 40 features based on posting behavior, friend network,

[1]Mostly taken from our published paper. **Rishabh Kaushal**, Chetna Sharma, and Ponnurangam Kumaraguru. Detection of Misbehaviors in Clone Identities on Online Social Networks. In *Proceedings of International Conference on Mining Intelligence and Knowledge Exploration, 2019.*

and profile attributes. We find that benign behaviors promote the celebrity they have cloned or seek attention, thereby helping in the celebrity popularity. However, on the contrary, we also find malicious behaviors (*misbehaviors*) wherein clone celebrities indulge in spreading indecent content, issuing advisories, and opinions on contentious topics. We evaluate our approach on a real social network (Twitter) by constructing a machine learning based model to classify behaviors of clone identities automatically and achieve accuracies of 86%, 95%, 74%, 92% & 63% for five clone behaviors corresponding to promotion, indecency, attention-seeking, advisory and opinionated.

7.1 Introduction

While in the real world, it is readily feasible to verify an individual's identity, it is quite tricky in OSNs [91]. The process of account creation is offered in quick and easy steps to encourage the adoption of OSNs platforms. This helps users create their accounts (also referred to as identities) with much ease. The user verification process is either bare minimal or non-existent at all. Consequently, the majority of the identities on the OSNs remain unverified. While it helps genuine users create identities easily, on the flip side, it also enables a malicious user to create identity *similar* to a genuine user (victim), which we refer to as *clone identity*[2] [143]. The public availability of Personally Identifiable Information (PII) of users, like, profile picture, bio details, and name makes the task of a malicious user even more trivial [144].

In this work, we focus our attention on celebrities' clone identities, referred to as *celebrity clones*. The motivations for a malicious user to create clone identities are many-fold, as exhibited by their behaviors. For instance, Fig 7.1 depicts victim (well known Indian film celebrity Amitabh Bachchan on Twitter, Fig 7.1(a)) along with his clone identity (Fig 7.1(b)), which has been in existence since 2009. Fan[3] identity (in case of celebrity) also exists as shown in Fig 7.1(c) along with an identity (Fig 7.1(d)), which has the same name but is neither clone nor fan. Celebrity clone identities indulge in several behaviors as depicted in Fig 7.2 such as promotion (Fig 7.2(a)), indecency (Fig 7.2(b)), attention-seeking (Fig 7.2(c)), advisory (Fig 7.2(d)) and opinionating (Fig 7.2(e)). In the case of celebrity cloning [16], the motivation is to ride on the popularity and reputation of known celebrities to influence users on OSN platforms. While behaviors associated with promotion and attention-seeking are benign, on the other hand, the behavior of spreading indecency is undoubtedly malicious. Also, the behaviors involving sending advisories and opinions, particularly on contentious issues, that misrepresent celebrities would be considered malicious behaviors. Besides celebrities, clone identities are being created for ordinary individuals as well, in order to create similar-looking profiles. These profiles are subsequently used to launch social engineering attacks like fake-following

[2]It is also referred as impersonation attack or identity clone attack.

[3]Fan identities are created by supporters of celebrities with benign intentions of popularizing the celebrity. Celebrities themselves may also create them, however, we do not delve into these issues, since our key focus is on the behavior of clone identities.

(a) Victim (b) Clone

(c) Fan (d) Other

Figure 7.1: Illustration of Victim, Clone, Fan and Other Identities in Twitter.

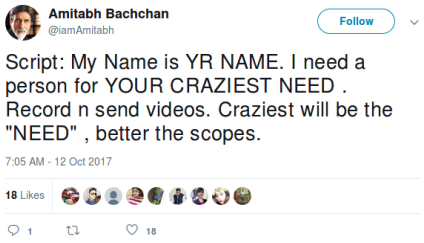
[2, 29], fake-likes [138], spear-phishing [121]. In this work, we do not consider clones of ordinary people since their reach and impact is mostly limited to the victim alone.

There are numerous fundamental challenges involved in our work.

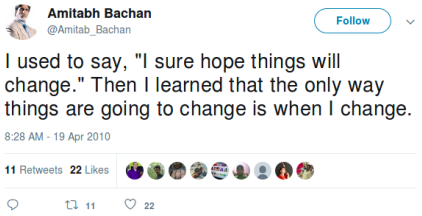
- The *first* challenge is to collect ground truth true verified identities of individuals over a given OSN platform. This is essential otherwise, there is no difference between real and clone identities.
- The *second* challenge is in *defining* clone identity, in other words, what are the user attributes (username, profile picture, or description) to be considered in order to say with a reasonable

(a) Promotion

(b) Indecency



(c) Attention Seeking



(d) Advisory

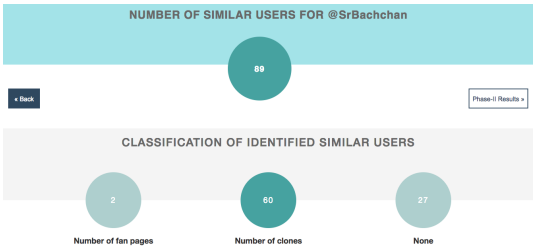
(e) Opinionated

Figure 7.2: Behavioral Characteristics Exhibited by Clone Identities.

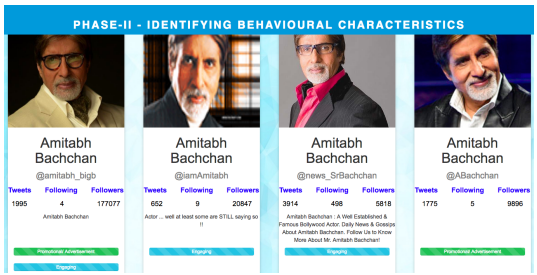
degree of confidence that a given identity is a clone of the victim.

- The *third* challenge is to perform *user search* within the OSN platform based on identified user attributes. Due to privacy issues, this feature has been deprecated in many platforms.

While clone detection and behavioral characterization are quite challenging, it is nevertheless a significant problem to be addressed. Misbehaviors of clone identities tarnish the reputation of their victim and can potentially involve them in unwarranted controversies. Therefore, it is essential to develop a solution using which celebrity can detect their cloned identities on a given OSN platform. Further, a celebrity would also want a solution to monitor the behaviors of their clone identities. To address these issues, we develop *CLONAWARE*, a web-based service that takes any Twitter handle as input (victim's true identity on Twitter), a snapshot of CLONAWARE is depicted in Fig 7.3, and outputs the number of fans, clones, and other similar identities to the victim. Additionally, the web



(a) Suspected Clone Identities classified into clones, fans or others (none) categories in CLONAWARE for a given input victim identity.



(b) Behavior of Clone Identities (Promotion, Indecency, Advisory, Attention & Opinionating) are displayed by CLONAWARE.

Figure 7.3: Work flow of CLONAWARE web service. Input is the Twitter handle of the victim (which in this case is Amitabh Bachchan @*SrBachchan*)

service also classifies the behaviors of clone identities into predefined classes, namely promotion, indecency, advisory, attention, and opinionating.

Our proposed solution of behavior characterization of clone identities consists of the following steps. In the first step, we find suspected clone identities of the victim. These suspected identities are marked as *clone* identities, *fan* identities (in case of celebrities), and *others* (also we use the term *none* interchangeably in this Chapter), as depicted in Fig 7.3(a). Model is trained on 1,614 identities using 13 features and achieves 86% accuracy with precision and recall of 88% & 83%, respectively, for detection of clones. In the second step, the behavioral characterization of each of the clone identity is performed into predefined categories based on their behavior, as shown in Fig 7.2. Five categories are considered namely promotion, indecency, attention-seeking, advisory, and opinionated. Our *behavioral characterization model*, pre-trained on 692 clones gives accuracies of 86%, 95%, 74%, 92% & 63%, respectively. CLONAWARE enables a victim to identify their clones and know undesirable behaviors of their clones, knowing which victim could initiate remedial measures (like reporting to OSN provider) to stay protected online.

Prior works have addressed the problem from the various standpoints. Clone identities come under a broader phenomenon of *fake identities* or *sybil identities* in which attacks may create identities that may not necessarily impersonate. To detect sybils, works [18, 155, 173] have leveraged network structure. Clone identities are detected by exploiting the fact that clones have similar attributes to real users [43, 71, 76]. However, most of these prior works have aimed to extract as many clone (or fake) identities as possible from a given OSN platform. Our approach is *user-centric* in the sense that we solve the problem for a specific victim through a web service, which we term as CLONAWARE, which would help a user remain *aware* of his/her clones and the behaviors of these clones. The key contributions of our work are as follows.

- Construction of clone detection model based on 13 features derived from a username, profile

description, user location, profile image, and URL mentioned by a user. The model analyzes over a hundred thousand tweets posted by 1,614 identities, of which 695 are clones, 134 fans, and the remaining 785 neither clones nor fans.

- Detailed characterization of clone behaviors into *five* categories, namely promotion, indecency, attention-seeking, advisory and opinionating, is performed. An exhaustive set of 40 features derived from content, network, and profile of 692 clones (and fans) identities are leveraged in the behavioral characterization model.

- Development of a web-based service, which we refer to as *CLONAWARE*, helps users remain aware of their clone identities. It takes any given victim identity on Twitter as input, identifies all clones (and fans in case of celebrity), and characterizes clone behavior among the five aforementioned behavioral types.

This chapter is organized as below. Section 7.2 gives a brief outline of the related work done by researchers in the field of fake identity detection, particularly in clone identity detection. Section 7.3 focuses on data collection and methodology for the creation of ground truth. Subsequently, we describe our proposed approach for the detection of clone identities and characterizing their behaviors in Section 7.4, followed by its evaluation and explanation of results in Section 7.5. Section 7.6 provides a detailed architectural description of our proposed web service CLONAWARE. Lastly, Section 7.7 highlights some key issues and limitations to our work.

7.2 Related Work

Clone identities are a specific case of fake identities in which the victim's PII are leveraged by an attacker to create real-looking identities. Detection of fake identities, referred to as Sybil attacks, are well studied. SybilGuard from Yu et al. [173] examined the impact of multiple fake identities (Sybil nodes) on honest nodes. Viswanath et al. [155] summarized the design of Sybil defense space from the perspective of detecting Sybils and tolerating (quantifying) their impact. Cao et al. [18] introduced a notion of ranking nodes (*SybilRank*) regarding their likelihood of being fake. While these works leverage network-based information in their solution approaches, Wang et al. [159] explored the possibility of a crowd-sourced solution for the detection of Sybils. Gupta et al. [49] leveraged the machine learning approach to detect fake accounts on Facebook.

In the context of clone detection, proposed solutions have exploited the fact that the attacker creates clone identities with attributes similar to that of the victim. Bilge et al. [11] demonstrated an identity theft attack on existing users of a given OSN and improved the trustworthiness of these identities by sending a friend request to friends of cloned victims. In another attack, they created cloned identities of victims across other OSNs where victims did not have their presence.

Jin et al. [71] exploited attribute similarity and common friends as critical indicators to find clone identities. Kharaji et al. [76] also explored the similarity of attributes and strength of relationships as essential features to detect clone identities. However, both [71] and [76] could not validate their proposed approach on the real OSN platform due to the unavailability of verified and their clone identities. He et al. [55] proposed a scheme to protect users from identity theft attacks. Gogo et al. [43] proposed a technique for the collection of impersonation attacks. Their findings suggest that these attacks target even ordinary individuals to create pseudo-real fake identities to evade detection.

7.3 Data Collection and Ground Truth

Among the various OSN platforms, we choose Twitter to evaluate our approach for many reasons. *First*, it is a popular short message service; users read and forward the tweets instantaneously. *Second*, it provides simple steps for account creation and has among the best support for developers, so creating a clone [40] is trivial. *Third*, Twitter follows a verification process for celebrities and grant a blue-colored verify badge[4] indicating verified account. Given that our data-driven machine learning-based approach is to detect clone identities automatically, data acquisition becomes an important step. Since the presence of clone identities targeting known celebrities are likely to be large, we use TwitterCounter[5], a web-based service to get 10,977 top influential (most followed) Twitter users spread across 227 countries. Due to computational constraints, we select the ten most followed (also referred as influential in this Chapter interchangeably) users[6] from India. For each of them, we perform user search on Twitter using Search API[7] using various combinations of the name of the user (first name only, the first letter of the first name + last name, both first name + last name and first name + the first letter of the last name). As a result, we obtain 1,614 suspected clone identities. We manually inspected each of these identities to determine whether they are indeed cloned identities or fan accounts (created to publicize or support their celebrities) or none of these. Out of 1,614 suspected clone identities, we find 695 to be clones, 134 fan identities, and the remaining 785 were neither clones nor fans, which forms ground truth for clone detection. Table 7.1 explains the breakup of these suspected clone identities. Given that the OSN platform, in this case, Twitter, provides excellent, one could presume that search would be sufficient to obtain clone identities, and hence, the need for a system like CLONAWARE is not required. However, on the contrary, it is quite evident that search results return almost 48% (785 out of 1,416) of users who are

[4]Verified Accounts on Twitter: https://help.twitter.com/en/managing-your-account/about-twitter-verified-accounts

[5]https://twittercounter.com/pages/100/

[6]Narendra Modi, Shah Rukh Khan, Amitabh Bachchan, Salman Khan, Akshay Kumar, Sachin Tendulkar, Virat Kohli, Deepika Padukone, Hrithik Roshan, and Aamir Khan

[7]Twitter Search API: https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets.html

Table 7.1: Distribution of Suspected Clone Identities into Three Categories namely Clones, Fans and Others

Victim Account	**Clones**	**Fans**	**Others**	**Total**
Narendra Modi (NM)	84	38	41	163
Shah Rukh Khan (SR)	56	11	41	108
Amitabh Bachchan (AB)	86	8	78	172
Salman Khan (SK)	23	7	42	72
Akshay Kumar (AK)	17	6	176	199
Sachin Tendulkar (ST)	107	10	70	187
Virat Kohli (VK)	79	30	20	129
Deepika Padukone (DP)	129	15	74	218
Hrithik Roshan (HR)	94	9	86	189
Aamir Khan (AAK)	20	0	157	177
Total	695	134	785	1,614

neither clones nor fans, thereby making search ineffective to detect clone identities. Furthermore, we observe that out of the total suspected clone identities for each influential user, almost half of them are clones except for *Akshay Kumar* and *Aamir Khan*. This is because the names of these two influential users are quite common among Indian people, and therefore we get a large number of identities, which are neither clones nor fans for them. It conforms to the findings of Perito et al. [123], which suggests that higher the uniqueness in the username, the more is the possibility of traceability. Further, we prepare ground truth for the behavior characterization of clones and fans.

Table 7.2: Distribution of Five Behavioral Categories (C1:Promotion, C2:Indecency, C3:Advisory, C4:Opinionating, C5:Attention) among Clones and Fans.

Victim Account	**C1**	**C2**	**C3**	**C4**	**C5**
Narendra Modi	8	9	7	61	27
Shah Rukh Khan	7	1	11	20	16
Amitabh Bachchan	14	3	12	28	29
Salman Khan	7	1	2	9	8
Akshay Kumar	5	0	1	12	5
Sachin Tendulkar	26	5	4	47	26
Virat Kohli	18	4	5	33	33
Deepika Padukone	27	12	10	52	42
Hrithik Roshan	19	7	9	30	28
Aamir Khan	6	0	1	6	6
Total	137	42	62	298	220

Out of 829 of these identities (695 clones and 134 fans), we found that 22 of them got suspended, and 115 of them did not post even a single tweet. So, ignoring these, we focused our attention on the remaining 692 identities by manually inspecting all the tweets posted by them and engagement

received. Based on the kind of content being posted, we narrowed down their behavior into *five behavioral categories*, namely promotion, indecent, advisory, opinions, and attention-seeking. The distribution of identities belonging to these categories are 137, 42, 62, 298, and 220, respectively as mentioned in Table 7.2. We observe that all these numbers add up to 759 which means that some of these identities exhibited more than one behavior.

7.4 Proposed Approach

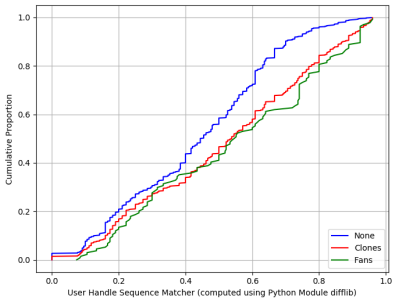
Given that we adopt a *user-centric* solution approach, so we provide the victim's user handle on Twitter as input in the first step. Various combinations of the victim's user name are used as keywords in Twitter Search API to obtain suspected clone identities in the *data collection* step, as discussed in Section 7.3. In the next step, pre-trained *clone detection* model is used to classify these suspected clone identities among *clones*, *fans* or *others* classes. Details of features employed by clone detection model are discussed in Section 7.4.1. Subsequently, we focus our attention on the behaviors exhibited by clones by performing behavioral characterization as discussed in Section 7.4.2.

7.4.1 Clone Detection

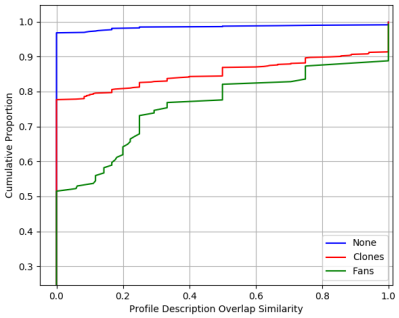
By definition, clone identities are those profiles which have *similar* appearance (profile attributes) with respect to their victim. Therefore, the key attributes that we consider for detecting clones are user handle, a profile description, user location, profile image, and URL mentioned by the user using which 13 features are derived. Table 7.3 lists various similarity features used for each of the attributes.

Table 7.3: Features used for Clone Detection (Total:13)

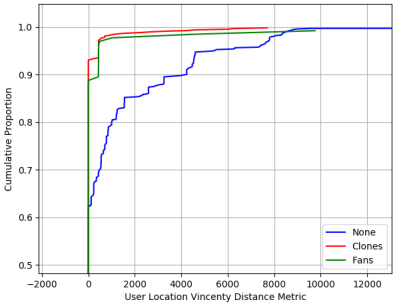
Attribute Name	**Similarity Features**
User Handle (2)	Sequence Matcher, Fuzzy Partial Ratio
Profile Description (4)	Matching Similarity, Jaccard Similarity, Overlap Similarity, Cosine Similarity
User Location (5)	Jaccard Similarity, Cosine Similarity, Overlap Similarity, Matching Similarity, Geo-Location
Profile Image (1)	Face++ Similarity
URL (1)	Exact Match



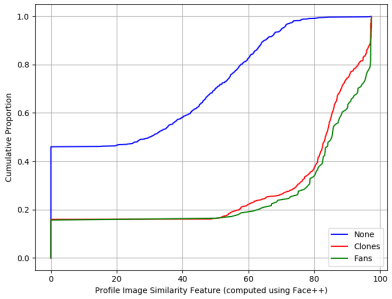
(a) Clones and Fans have higher degree user handle matching with victim.



(b) Clones and Fans have more similar profile description with victim.



(c) Clones and Fans have higher likelihood of belonging to same location (less distance) as victim.



(d) Face++ image similarity is higher for clones and fans with that of victim.

Figure 7.4: CDF graphs for clone detection features.

- **User Handle**: Users on Twitter create their user handles (or usernames). Clones are most likely to generate user handles that are quite similar to their victims. We have used two similarity metrics to compare user handles, namely *sequence matcher* and *fuzzy partial ratio*. SequenceMatcher[8] is a class provided by $difflib$ python module which finds the longest contiguous subsequence between two input strings. Fuzzy partial ratio is provided by FuzzyWuzzy[9], an open-source fuzzy string matching python module which compares two strings within the best matching length, so that string length does not affect the outcomes adversely. Fig 7.4(a) clearly depicts higher user handle match for clone and fan with victim.

- **Profile Description**: Clones imitate or claim to be the real account of a victim, and in the process, they end up having their profile descriptions quite similar to that of their victim. We employ four similarity measures. *Matching similarity* computes intersecting words. *Jaccard similarity* treats description text as a set of words and normalizes intersecting words with total unique words belonging to both strings. *Overlap similarity* divides the number of common words with the length of shorter string among the two strings being compared. Fig 7.4(b) shows that clones and fans have their profile descriptions quite similar to the victim. *Cosine similarity* converts two strings into vectors and computes cosine between them.

- **User Location**: Most of the clones have specified the same location as that of their victim. In addition to the similarity metrics used for profile description, we employed GeoPy[10] to find latitude and longitude for a given location. Subsequently, we use *vincenty* based geodesic distance to compute the distance between clone and victim profiles. Distances are found to be extremely less, as depicted in Fig 7.4(c) between clones and their victim.

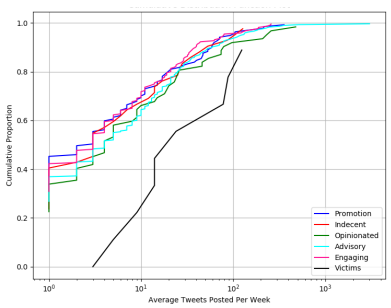
- **Profile Image**: Out of the above, this is the most important feature used by clones to impersonate victims. Naive users would easily get tricked into believing the clone profile to be that of the victim's profile. We leverage Face++[11] to compare the faces in two given images from clone and victim profiles. Face++ API returns a confidence score between 0 to 100, which we use as an image similarity feature. As depicted in Fig 7.4(d), more than 50% of clones and fans have very high image similarity (80% & above) with the victim.

- **URL**: Lastly, there is an option for users to enter URL on their profile. We perform an exact match on the URL specified in clone and victim profiles so that this feature is 1 or 0 depending upon whether an exact match was found or not.

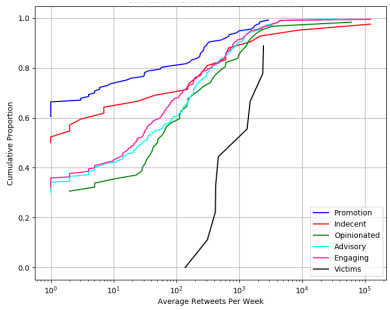
[8] https://docs.python.org/2/library/difflib.html
[9] https://github.com/seatgeek/fuzzywuzzy
[10] Geocoding service client: https://pypi.org/project/geopy/
[11] https://www.faceplusplus.com/

7.4.2 Behavioral Characterization

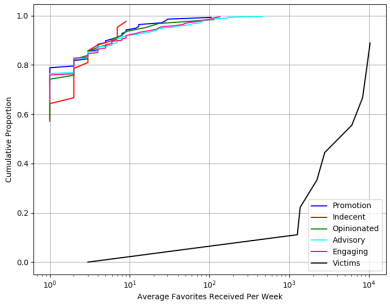
Once we have detected clones, as explained in data collection, the next step is to characterize their behavior. There are *five* behavioral categories that we focus upon, namely promotion, indecent, opinionated, advisory, and attention-seeking. During our behavioral characterization study of clones, as depicted in Fig 7.5, we found that clones exhibit lessor activity weekly in terms of tweets posted (Fig 7.5(a)) and tweets retweeted (Fig 7.5(b)) , and favorites (Fig 7.5(c)) received as compared to victims who are influential users on Twitter.



(a) Average Tweets Posted Per Week



(b) Avg. Retweets Received Per Week



(c) Avg. Favorites Received Per Week

Figure 7.5: Behavioral Characteristics Exhibited by Clone Identities.

Table 7.4 describes the details of 40 features employed for behavioral characterization. We compute each of the features marked with '*' weekly, and we consider minimum, maximum, average, and standard deviation for each of them as features. We divide features into three categories: content, network, and profile, depending upon the type of attribute used as the source for feature computation.

- **Content Based Features**: The kind of content posted by clones provides a good indication of the type of behavior exhibited. The presence of URLs could lead users to inappropriate sites or promotional content. For instance, promotional keywords [5] would indicate promotion (or advertisement) class. Currency symbols could attract users towards some promotion. The

Table 7.4: Features for Behavioral Characterization

Features Type	**List of Features**
Content based Features (21)	URLs, Promotional Keywords, Mentions, Currency Symbols, Question Marks, Engaging Words, Swear Words, Quotes, Advisory Keywords, Days Since Last Tweet, Time* between Two Tweets, Tweet* Length, Exclamation, Colon-Semicolon.
Network based Features (14)	Tweets* per week, Retweet* Count & Favorite* Count, Followers, Following.
Profile based Features (5)	Bio Analysis - URLs, Length, Victim Tag, Fan or Clone, Mention, Handle Mention.

presence of question marks and engaging words (like *who*, *what*, *when*, and*where*) could be used to invite attention or engagement. Swear words [108] would indicate the presence of indecency. Special characters like quotes and advisory keywords (like *should*, *said*, and *quote*) could indicate self-help or advisory. Besides these, we use generic features like hashtags, tweet length, the time between two tweets, days since the last tweet, presence of exclamation symbol, and colon-semicolon.

- **Network Based Features**: We study the behavior of clone identities with their ego network by measuring the engagement. Therefore, we compute features like retweet count, favorite count, tweets per week, number of followers, and the following in network-based features.

- **Profile Based Features**: Twitter has very few profile attributes, among which *user bio* is worth investigating. We compute the number of occurrences of URLs, victim names (or tag) along with the length of bio in the user bio-field as features. Also, to capture the nature of the profile, we look into the occurrence of common words. A clone may use words like *real account* or *official account*, whereas a fan page bio may have *unofficial page*, *parody account*, or *fan association* mentioned.

During our behavioral characterization study of clones, as depicted in Fig 7.5, we found that clones exhibit lessor activity on a weekly basis in terms of tweets posted (Fig 7.5(a)), tweets retweeted (Fig 7.5(b)) and favorites (Fig 7.5(c)) received as compared to victims who are influential users on Twitter. While this is true, it ought not to be assumed to be always valid in the case of an ordinary individual as victims.

7.5 Evaluation and Results

We explain our evaluation methodology and corresponding results obtained for both clone detection and behavioral characterization.

7.5.1 Evaluating Clone Detection Model

In this section, we use 695 clones, 134 fans, and 785 others, as the ground truth and answer the following.

- Which is the best classifier for clone detection?
- What is the detection accuracy when the problem is *recast* as binary classification?
- Which user attribute has a maximum impact on detection accuracy?
- Is the learned model generic enough to be applied to any victim ?

Identifying Best Classifier

To identify the best classifier for clone detection, we fed our dataset comprising of 13 features for 1,614 suspected clone identities into various machine learning algorithms listed in Table 7.5. 10-fold cross-validation is used along with 80:20 training-test split in all runs of all classifiers. It turns out

Table 7.5: List of Classifiers along with their Accuracies

Name of Classifier	**Accuracy**
RandomForestClassifier	**0.80**
DecisionTreeClassifier	0.78
LogisticRegression	0.77
KNeighborsClassifier	0.77
ExtraTreesClassifier(ensemble)	0.76
LogisticRegressionCV	0.76
RidgeClassifierCV	0.75
RidgeClassifier	0.74
ExtraTreeClassifier(tree)	0.72
Neural Network - MLPClassifier	0.71
LinearSVC	0.69
Naive Bayes-BernoulliNB	0.56
Naive Bayes-GaussianNB	0.54

that *random forest classifier* performs the best with accuracy of 80%. Furthermore, as shown in

Table 7.6, the best classifier is able to achieve higher precision and recall for *clone* class and *others* class.

Binary Classifier Performance

Results are not good for *fan* class due to less training data because out of 1,614 identities, only 134 fans were present. Given that in most of the scenario, when the victim is an ordinary individual,

Table 7.6: Precision, Recall and F1 Score for Best Classifier.

Three Class Classification			
Metric / Class	**Clones**	**Fans**	**Others**
Precision	0.77	0.43	0.86
Recall	0.79	0.29	0.88
F1-Score	0.78	0.35	0.87
Two Class Classification			
Metric / Class	**Clones**	**Others**	
Precision	0.88	0.84	
Recall	0.83	0.89	
F1-Score	0.85	0.87	

fan class do not exist, we also experimented by removing *fan* class and found that precision and recall increases for both *clone* and *others* classes in binary classification settings as depicted in Table 7.6. We achieve an accuracy of 86% in binary classification settings with the best precision and recall of 88% and 83% for clones.

Attribute Importance

In this evaluation, we study the importance of attributes in the clone detection model. Recall from Table 7.3 of features employed for the clone, we find that five attributes were used, namely user handle, a profile description, user location, profile image, and URL. To study the impact of these attributes, we remove attributes (one by one) and all features derived from it while doing model training and measure classification performance, as shown in Fig 7.6. Classifier performance is only marginally affected when we remove profiles like attributes user handle, a profile description, user location, and URL mentioned by the user. However, when the profile image attribute is removed, performance significantly reduces, thereby indicating that the profile image attribute is most important in the clone detection model.

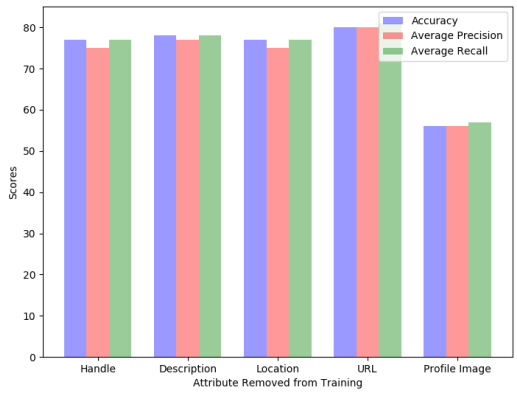


Figure 7.6: Measuring impact of attribute on classifier performance. Removing profile image attribute reduces accuracy, precision & recall significantly, thereby indicating that profile image attribute is the most important in clone detection.

Generic Applicability Evaluation

In any data-driven solution, a genuine concern is whether the learned model is robust enough and capable of being applied in a generic scenario. In our case, recall from Table 7.1 that we have trained our clone detection model using the ten influential users (celebrities) as the seed. To test the generic applicability, we train the best classifier (random forest) on identities, which are either clones, fans, or others for 9 of these influential users and test the identities: either clones, fans, or others of the remaining influential user.

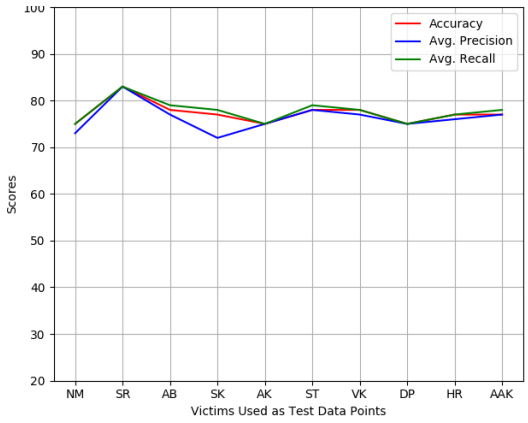


Figure 7.7: Evaluating generic applicability of clone detection model. X-axis represent the victim whose clones, fans & others identities are excluded from training and instead used exclusively as test data. Shorthand notations used represent the first letters of first and last names of the influential users as listed in Table 1 (eg. NM means Narendra Modi).

As depicted in Fig 7.7, the performance measured in terms of accuracy, precision, and recall remains consistently hovering around the range of 72% to 83% with an average of 77% and 0.026 standard

deviation which is quite less.

7.5.2 Behavioral Characterization

Recall from Table 7.2 that 692 clones (and fan) identities were analyzed to categorize them into one (or more) of the behavioral types. In particular, 137 were found to be involved in the promotion, 42 in spreading indecency, 64 in advisory, 298 in opinionating, and 220 in attention-seeking. We use this as ground truth and answer the following research questions (RQs).

- RQ1: Which is the best classifier for behavior characterization of clones?
- RQ2: Does detection accuracy improve with more training?

Identifying Best Classifier

To identify the best classifier, we compute 40 features on the 692 identities and ran over 12 off-the-shelf classifiers namely Random Forest, Decision Tree, Logistic Regression, KNeighbors, ExtraTreesClassifier, Logistic Regression, Ridge Classifier, ExtraTree Classifier, Neural Network - MLPClassifier, LinearSVC and Naive Bayes Classifier (Bernoulli and Gaussian). In our experimental set-up, we consider the multi-class (five classes) problem as five different binary classification problems in which the goal is to detect the presence or absence of a specific behavior in a given clone identity. It turns out that there is no single classifier, which performs best for all behavior types. Random forest works the best (94%) for detecting indecency, Naive-Bayes detects promotion with 86% accuracy, Logistic Regression gives 74% accuracy for attention-seeking behavior, RidgeClassifier gives 92% accuracy for advisory behavior whereas ExtraTreesClassifier gives 63% accuracy for opinionated content spreading. The difference in accuracy values is due to the difference in the amount of labeled data available for these classes. Besides training data size, the distinguishing features in some classes appear more prominently than other classes. For instance, to detect 'indecency', swear words are an important distinguishing feature than for other classes, say 'advisory.'

Training-Testing Split

In this evaluation, we study the effect of train-test split on classifier performance. As evident from Table 7.8, the classification accuracy is improved in all behavioral types as we increase the train-test ratio from 50-50 to 80-20, which suggests that as training size would size, the accuracies will improve. Also, we observe that the accuracy of the advisory class is low due to less number of clones spreading advisory behavior (Table 7.2). On the contrary, the indecent class's accuracy is

Table 7.7: List of Classifiers along with their Accuracies for five kinds of behaviors namely Promotion (Pr), Indecency (In), Attention Seeking (At), Advisory (Ad), and Opinionated (Op)

Name of Classifier/Class	**Pr**	**In**	**At**	**Ad**	**Op**
RandomForestClassifier	0.79	**0.94**	0.69	0.88	0.57
DecisionTreeClassifier	0.73	0.88	0.65	0.80	0.55
LogisticRegression	0.78	0.92	0.67	0.87	0.52
KNeighborsClassifier	0.91	0.94	0.66	0.84	0.52
ExtraTreesClassifier(ensemble)	0.76	0.94	0.61	0.87	0.52
LogisticRegressionCV	0.76	0.92	**0.74**	0.89	0.49
RidgeClassifier	0.79	0.93	0.64	**0.92**	0.55
ExtraTreesClassifier(tree)	0.73	0.90	0.64	0.84	**0.63**
Neural Network - MLPClassifier	0.70	0.89	0.59	0.87	0.39
LinearSVC	0.76	0.91	0.58	0.75	0.45
Naive Bayes-BernoulliNB	**0.86**	0.91	0.46	0.89	0.54
Naive Bayes-GaussianNB	0.33	0.26	0.35	0.25	0.47

Table 7.8: Accuracy scores with different training-testing split for five kinds of behaviors namely Promotion (Pr), Indecency (In), Attention Seeking (At), Advisory (Ad), and Opinionated (Op)

Train-Test	**Pr**	**In**	**At**	**Ad**	**Op**
80-20	**0.86**	**0.94**	**0.92**	**0.63**	**0.74**
70-30	0.73	0.94	0.90	0.56	0.68
60-40	0.82	0.91	0.90	0.54	0.61
50-50	0.80	0.92	0.90	0.54	0.65

high, even though the number of indecent instances is less. We attribute it to the fact that swear words in indecency are limited and highly discriminative.

7.6 CLONAWARE

In this section, we give a detailed description of CLONAWARE, our proposed web-service for detection and behavioral classification of clones. Given that CLONAWARE web-service is powered by these learning models at the back-end, therefore, it can be used by any user, both celebrity or ordinary individual.

7.6.1 System Design

CLONAWARE web-service is based on Flask[12], which is a Python micro web framework. The detailed control flow of CLONAWARE is depicted in Fig 7.8, which explains the steps a user would

[12]Flask: http://flask.pocoo.org/

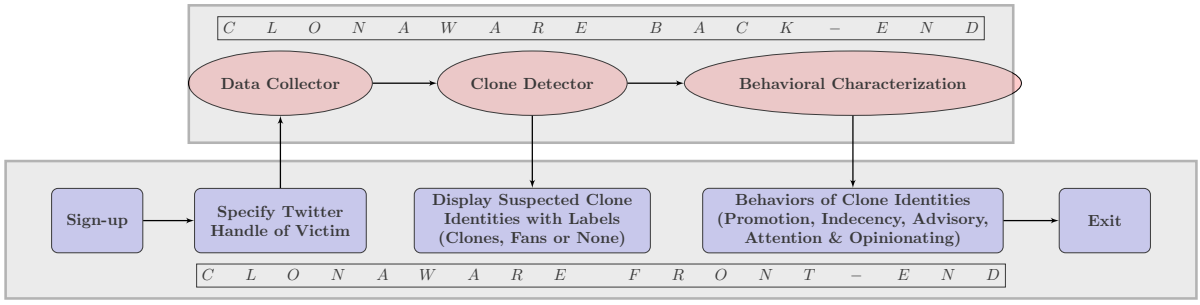


Figure 7.8: Architecture Diagram of CLONAWARE web service.

typically follow to use the service as enlisted below.

1. User signs-up and logs into the web-service.
2. User enters the Twitter handle of the victim for whom analysis is to be performed.
3. Data Collection module is invoked, which uses various username combinations to obtain a list of suspected clone identities.
4. The clone detection features (Table 7.3) are computed for these identities, which are subsequently, fed into the *clone detection model* which is pre-trained on 1,614 identities.
5. Output of clone detection model is displayed to the user (Fig 7.3(a) shown earlier in this Chapter) in which the number of clones, fans, and other identities can be seen.
6. Features for behavioral characterization of clones (Table 7.4) are computed which are fed into *behavioral characterization model* as input.
7. Output of behavior characterization of clones are displayed to the user (Fig 7.3(b) shown earlier in this Chapter) in which type of behavior for each clone is specified.

7.7 Discussion and Limitation

In this chapter, we apply a solution to the user identity linkage across social networks to solve the problem of identity clones. We *recast* the problem as a binary classification problem, and conventional classifiers are applied and empirically evaluated. For the clone detection, we evaluate using 13 features computed from the profiles of 1,614 identities and achieve an accuracy of 86% in binary classification settings (clones and others) with the best precision and recall of 88% and 83%, respectively. We extract an exhaustive set of features from network, content, and profile of celebrity clone identities. Best classifiers achieve accuracies of 86%, 95%, 74%, 92%, and 63% for

five clone behaviors, namely promotion, indecency, attention-seeking, advisory, and opinionated, respectively. We develop CLONAWARE web service using which any user can find his/her clones and become aware of their clones' behaviors on Twitter. Once users become *clone-aware*, they can take remedial measures of reporting to Twitter to protect their identity.

There are a few limitations to this work. We carefully select Twitter as the social network platform because it provides a mechanism of *verified accounts* in which a blue tick appears in the user profile. This helped us to identify the real account from the cloned identities correctly. It will be hard to obtain ground truth in social networks that do not have any in-built mechanism for verification Owing to computation limitations, we restrict ourselves to suspected 1,614 clones of the top ten celebrities on Twitter only from India. Therefore, we have a limited and biased dataset. It would be nice to extend the work on celebrities in other countries to understand the influence of cultural factors on clone behaviors. We conveniently selected celebrities as victims because ground truth for them is readily available, and they have more clones than ordinary persons. Lastly, while the accuracies of behavioral prediction of promotion (86%), indecent (95%) and advisory (92%) are quite decent, at the same time, the accuracies for categories like attention (74%) and opinions (63%) are way too less to be of practical use. More data needs to be collected to improve accuracies for predicting these behaviors. This work can also be extended to build an application that alerts celebrities whenever any clone indulges in any misbehavior. We understand that every celebrity would have a public relations team, who can benefit from such an application.

Chapter 8

Conclusion, Limitation & Future Work

In this chapter, we summarize our work's main contributions in addressing user identity linkage at different levels. Subsequently, we highlight the limitations of work and present directions for future work.

8.1 Summary of Contributions

The main contribution of this thesis is to address the problem of user identity linkage from different perspectives. From a data collection standpoint, we started with a comparative study of several data collection methods to gather linked user identities. For the fairness of the dataset, we observed and investigated the inherent biases in the identity linkage datasets. At the level of the proposed approach, we proposed a novel NexLink framework that leverages network embeddings to extract cross-network linkages between two social networks. From the user's privacy perspective, we developed a system that leverages the concept of soft paternalist nudges to inform the linkability of users' identities across social networks and helped them control linkability. Lastly, we proposed a machine learning-driven methodology to detect identity clones within the same network. Next, we explain these contributions in detail.

Analysis of Data Collection Methods. An important first step is to collect user accounts (identities) belonging to the same person across social networks to solve the user identity linkage problem. To this end, we perform a systematic comparative study of five methods, which we refer to as Advanced Search Operator (ASO), Social Aggregator (SA), Cross-Platform Sharing (CPS), Self-Disclosure (SD) and Friend Finding Feature (FFF). Taking all these methods together, we

collect linked identities belonging to 208,120 individuals across 43 different OSNs. We compare data collection methods quantitatively based on social network coverage and the number of linked identities obtained per-individual. We also perform a qualitative assessment of these data collection methods based on completeness, validity, consistency, accuracy, and timeliness.

Investigation of Biases in Identity Linkage DataSets. On observing the data collection methods, we find that they leverage user behaviors on different social networks to collect linked identities. As a consequence, user behavioral biases get manifested in the datasets thus obtained. Therefore, we perform a detailed investigation into the dataset biases, a work that had mostly remained underexplored in the user identity linkage research. More specifically, we characterize, detect, and quantify biases in these datasets. We find that biases manifest in the form of lexical differences in user-generated content, particularly in usernames and display names configured by users. For quantification, we employ a measurement approach, referred to as *situational testing*, which is used in discrimination studies, and adapt it to quantify biases in user identity datasets.

NeXLink: Node Embedding Framework for Cross-Network Linkages. After analyzing data collection methods and investigating dataset biases, we propose *NeXLink*, a modular and flexible node embedding framework for cross-network linkages (CNLs), a pair of user identities across two different social networks belonging to the same individual. For developing a node embedding framework, we model the social network as a graph. We optimize our node embeddings by ensuring that users belonging to CNLs are closer in embedding space than other nodes, using only the network information. Our NeXLink framework comprises of three steps. First, we obtain local node embeddings by preserving the local structure of nodes within the same social network. Second, we learn the global node embeddings by preserving the global structure, which is present in the form of common friendship exhibited by nodes involved in CNLs across social networks. Third, we combine the local and global node embeddings, which preserve local and global structures to facilitate the detection of CNLs across social networks. We evaluate our proposed framework on an augmented (synthetically generated) dataset of 63,713 nodes & 817,090 edges and real-world dataset of 3,338 Twitter-Foursquare node pairs. Our approach achieves an average hit rate of 98% and 88% in augmented and real-word dataset, respectively, for detecting CNLs across social networks and significantly outperforms previous state-of-the-art methods.

Nudging Nemo: Helping Users Control Linkability across Social Networks. Linkage of user identities across social networks collapses the user context and has privacy implications, particularly for those users who do not want their identities to be linkable across networks. Therefore, we propose and develop *Nudging Nemo*, a system that helps users to control the linkability of their identities across multiple platforms. It has two components, namely a linkability calculator and a

soft paternalistic nudge. Linkability calculator uses state-of-the-art identity resolution techniques to compute a normalized linkability measure for each pair of social network platforms used by a user. Soft paternalistic nudge alerts the user if any of their activity violates their preferred linkability. We evaluate the effectiveness of the nudge by conducting a controlled user study on privacy-conscious users who maintain their accounts on Facebook, Twitter, and Instagram. Outcomes of user study confirm that the proposed framework helped most of the participants to make informed decisions, thereby preventing inadvertent exposure of their personal information across social network services.

Detecting of Clone Identities in Online Social Networks. In this last contribution, we present user identity linkage methods to detect similar-looking user identities within the same social network, which we refer to as clone identities. We focus on clone identities of famous persons (celebrity clones) who ride on the credibility and popularity of celebrities to gain engagement, impact, and at times indulge in malicious activities. We build an automated clone detection model to identify the clones of a given victim. This approach is quite similar to solution approaches for user identity linkage, the difference being that we are detecting identities exhibiting similar features within the same social network. We evaluate our clone identity detection approach on 1,614 identities using 13 features and achieve 86% accuracy with precision and recall of 88% & 83%, respectively. Next, we build a model that automatically characterizes the behavior of clone identities into five categories based on the content being posted by them: promotion, indecency, attention-seeking, advisory, and opinionated. To this end, we extract an exhaustive set of 40 features based on posting behavior, friend network, and profile attributes. We evaluate our approach on a real social network (Twitter) by constructing a machine learning based model to classify behaviors of clone identities automatically and achieve accuracies of 86%, 95%, 74%, 92% & 63% for the five clone behaviors corresponding to the promotion, indecency, attention-seeking, advisory and opinionated.

8.2 Limitations

In this section, we discuss the limitations and challenges encountered during the conduct of work presented in this thesis based on our experiences.

8.2.1 Collection of linked user identities

Data collection, particularly the users' data, is a challenge considering the privacy issues surrounding users' data. Therefore, we observe that social media platforms' API support has been dwindling in their capabilities. Twitter, by far, has the most supportive API, and a lot of users' data can be collected. However, the same cannot be said for other social media platforms like Facebook

and Instagram. Consequently, in our work of implementing data collection methods to gather linked user identities as discussed in Chapter 3, particularly cross-platform sharing (CPS), we keep Twitter as a platform where users cross-post, so that we can search through tweets based on keywords. This could be possible because of the post search capability provided by the Twitter API. The same could not be possible on Facebook. Besides API dependency, since most data collection methods depend upon user behavior, the amount of data that can be gathered is proportional to the data shared by users in the public domain. For instance, in social aggregation (SA) websites like About.me, we can only get as much linked user identities as shared by the user. The same limitation holds for self-disclosure (SD) method, where the users themselves put their identities on other social media platforms. Lastly, some of the data collection methods employ web crawling to gather content shared in the public domain by the user. All the content that we obtain is for the use of academic research only. We are limited by the rate limits and rules specified in the *robot.txt* file on what pages on a website can be crawled.

8.2.2 Linked user identity Dataset Biases

We leveraged users' behaviors to collect linked identities across different social media platforms. Consequently, user behavior biases also get manifested in the user identity datasets. In Chapter 4, we focussed on the cross-platform sharing (CPS) and self-disclosure (SD) method driven datasets. To characterize the biases in these datasets, we restricted ourselves to studying only username and display name being configured by users as users' behaviors. In social media platforms, namely Twitter and Instagram, which we studied, users can change their usernames and display names. In terms of user behaviors, we limit ourselves to only user behaviors which change username and display name.

8.2.3 Linkage of user identities

Following the latest trend of leveraging node embedding approaches for learning graph representations, we proposed the NexLink framework in Chapter 5. Our framework is modular and flexible because it allows the usage of different embedding approaches for constructing local and global node embeddings. Our proposed NeXLink framework is restricted to use only network-related information in a given social network modeled as a graph.

8.2.4 Challenges and Improvements in Nudge

The purpose of linkability nudge, discussed in Chapter 6, is to help users understand the factors contributing to the linkability of their social media identities. Further, it nudges the user every time the user performs a behavior (like making a post or making a change in his/her profile settings),

which results in driving linkability score away from their desired linkability score. The linkability nudge is implemented as a browser plugin, and it keeps track of user changes on the specific pages of social media websites (like Facebook's profile page and Twitter's tweet posting page). Therefore, to keep the nudge working, it needs a development team that periodically keeps updating the tracking code each time the social media page makes changes in HTML elements. Further, to operate, the linkability nudge (app) is required to be registered as a developer's application on social media platforms (Facebook and Twitter, in our case). After the issue of Cambridge Analytica involving Facebook [12], our application was denied authorization to collect user's access token on Facebook. For the evaluation of our linkability nudge, we limited ourselves to the conduct of a controlled lab study based evaluation.

8.2.5 Applications of User Identity Linkage

Several applications would stand to benefit from the solution of the user identity linkage problem. One of the applications that we discussed in Chapter 7 is to detect clone identities. We could not deploy the web service CLONAWARE in the public domain. An extension would be to perform a detailed usability study to ascertain the benefits and limitations of a web service that detects clones and alerts about misbehavior performed by clone identities.

8.3 Future Work

We discuss future directions in this section. In terms of data collection of linked identities, there can be two broad directions to move forward. The first direction is discovering how users' identities can be linked by observing users' behaviors. For instance, on LinkedIn, occasionally, there are job-related posts that ask for users to put their PII (like mobile numbers or email) as comments so that more details about the job can be shared. This PII can search for these users' identities on social media platforms that support PII-based user searches. So, in the way, users' identities on LinkedIn and other social media platforms can be linked. The second direction is continuing with the same methods as discussed in Chapter 3 but exploring other websites and social media platforms. For instance, we worked on About.me for social aggregation method, other similar websites like *linktr.ee*[1] can be explored. Similarly, the collection of linked user identities on other less explored social media platforms like strava[2] (app for cyclists and runners) and goodreads[3] (for book recommendations) can be tried.

In the context of dataset biases, other ways of ascertaining user behaviors like posts made by users

[1] https://linktr.ee/
[2] https://www.strava.com/
[3] https://www.strava.com/

and friends of users need to be explored, because biases induced by these other behaviors may also be present. In quantifying the extent of biases in identity linkage datasets, we applied the concept of situational testing, which is usually employed in the discrimination studies. Extending the idea further, we can perform detection, quantification and prevention by drawing more ideas from bias studies [44, 125], discrimination studies [52, 131] and fairness preserving algorithmic studies [20, 86, 174, 177].

For learning node embeddings, we can also include node attributes derived from user profile configuration and user content in the NeXLink framework and study their impact on performance measured. One recent work from Hadgu et al. [50] proposes an approach that jointly models heterogeneous data, namely profile name, images, text content, and network structure to link user identities on Twitter and DBLP. For optimization of node embeddings, we employ a random walk based approach. In this context, another direction would be to explore deep learning-based methods for optimizing node embedding vectors.

Future developments of nudges shall have to consider the limitation of access token authorization granted by the social media platform. As one direction in future, we can perform a field trial of linkability nudge by offering it as a web service for any online user to use it and benefit. It can be poised as a service to help users reduce the linkability of their social media identities, or even a service to alert users about the offensive posts they would have inadvertently made in the past.

Besides above, there are more use cases, discussed earlier in Section 1.1, where benefits from user identity linkages can be explored further. For instance, one direction could be to perform a behavioral analysis of developers. During the recruitment process in software companies, it has become common for recruiters to look at GitHub profiles of prospective developers. Also, if these developers' participation in StackOverflow, can also be obtained, it can provide more insights into their behavioral aspects like helpfulness, respectfulness, breadth, and depth of expertise. Similarly, other applications can be explored in the future.

Bibliography

[1] Alessandro Acquisti, Idris Adjerid, Rebecca Balebako, Laura Brandimarte, Lorrie Faith Cranor, Saranga Komanduri, Pedro Giovanni Leon, Norman Sadeh, Florian Schaub, Manya Sleeper, et al. Nudges for privacy and security: Understanding and assisting users' choices online. *ACM Computing Surveys (CSUR)*, 50(3):1–41, 2017.

[2] Anupama Aggarwal, Saravana Kumar, Kushagra Bhargava, and Ponnurangam Kumaraguru. The follower count fallacy: detecting twitter users with manipulated follower count. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, pages 1748–1755, 2018.

[3] Mishari Almishari, Dali Kaafar, Ekin Oguz, and Gene Tsudik. Stylometric linkability of tweets. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, pages 205–208. ACM, 2014.

[4] Hazim Almuhimedi, Florian Schaub, Norman Sadeh, Idris Adjerid, Alessandro Acquisti, Joshua Gluck, Lorrie Faith Cranor, and Yuvraj Agarwal. Your location has been shared 5,398 times! a field study on mobile app privacy nudging. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pages 787–796, 2015.

[5] Contributing Author. Magic marketing words you should be using. *Vertical Response*, September 2017. [Online; posted 19-September-2017].

[6] Albert Bandura. Social cognitive theory of personality. *Handbook of personality*, 2:154–96, 1999.

[7] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[8] Sergey Bartunov, Anton Korshunov, Seung-Taek Park, Wonho Ryu, and Hyungdong Lee. Joint link-attribute user identity resolution in online social networks. In *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining, Workshop on Social Network Mining and Analysis*. ACM, 2012.

[9] Joseph B Bayer, Nicole B Ellison, Sarita Y Schoenebeck, and Emily B Falk. Sharing the small moments: ephemeral social interaction on snapchat. *Information, Communication & Society*, 19(7):956–977, 2016.

[10] Nacéra Bennacer, Coriane Nana Jipmo, Antonio Penta, and Gianluca Quercini. Matching user profiles across social networks. In *International Conference on Advanced Information Systems Engineering*, pages 424–438. Springer, 2014.

[11] Leyla Bilge, Thorsten Strufe, Davide Balzarotti, and Engin Kirda. All your contacts are belong to us: automated identity theft attacks on social networks. In *Proceedings of the 18th international conference on World wide web*, pages 551–560. ACM, 2009.

[12] Carissa Boerboom. Cambridge analytica: The scandal on data privacy. 2020.

[13] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. 1998.

[14] Francesco Buccafurri, Gianluca Lax, Antonino Nocera, and Domenico Ursino. Discovering links among social networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 467–482. Springer, 2012.

[15] Chase Buckle. Globalwebindex, April 2018. [Online; posted 11-April-2018].

[16] Madeline Buxton. The social scam: For a-listers, imposters still loom large. *Refinery29*, May 2018. [Online; posted 2-May-2018].

[17] Toon Calders and Indrė Žliobaitė. Why unbiased computational processes can lead to discriminative decision procedures. In *Discrimination and privacy in the information society*, pages 43–57. Springer, 2013.

[18] Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, pages 15–15. USENIX Association, 2012.

[19] Francesca Carmagnola, Francesco Osborne, and Ilaria Torre. User data distributed on the social web: how to identify users on different social systems and collecting data about them. In *Proceedings of the 1st international workshop on information heterogeneity and fusion in recommender systems*, pages 9–15, 2010.

[20] Carlos Castillo. Fairness and transparency in ranking. In *ACM SIGIR Forum*, volume 52, pages 64–71. ACM, 2019.

[21] Hongxu Chen, Hongzhi Yin, Weiqing Wang, Hao Wang, Quoc Viet Hung Nguyen, and Xue Li. Pme: projected metric embedding on heterogeneous networks for link prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1177–1186, 2018.

[22] Siyuan Chen, Jiahai Wang, Xin Du, and Yanqing Hu. A novel framework with information fusion and neighborhood enhancement for user identity linkage. *arXiv preprint arXiv:2003.07122*, 2020.

[23] Wei Chen, Hongzhi Yin, Weiqing Wang, Lei Zhao, Wen Hua, and Xiaofang Zhou. Exploiting spatio-temporal user behaviors for user linkage. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 517–526. ACM, 2017.

[24] Anfeng Cheng, Chun-Yi Liu, Chuan Zhou, Jianlong Tan, and Li Guo. User alignment via structural interaction and propagation. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[25] Xiaokai Chu, Xinxin Fan, Di Yao, Zhihua Zhu, Jianhui Huang, and Jingping Bi. Cross-network embedding for multi-network alignment. In *The World Wide Web Conference*, pages 273–284, 2019.

[26] J. Clement. Number of social media accounts (2019), January 2019. [Online; posted January-2019].

[27] Denzil Correa, Leandro Araújo Silva, Mainack Mondal, Fabrício Benevenuto, and Krishna P Gummadi. The many shades of anonymity: Characterizing anonymous social media content. In *Ninth International AAAI Conference on Web and Social Media*, 2015.

[28] Denzil Correa, Ashish Sureka, and Raghav Sethi. Whacky!-what anyone could know about you from twitter. In *2012 Tenth Annual International Conference on Privacy, Security and Trust*, pages 43–50. IEEE, 2012.

[29] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. Fame for sale: efficient detection of fake twitter followers. *Decision Support Systems*, 80:56–71, 2015.

[30] Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. A survey on network embedding. *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[31] Jenny L Davis and Nathan Jurgenson. Context collapse: Theorizing context collusions and collisions. *Information, communication & society*, 17(4):476–485, 2014.

[32] Li Deng. A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Transactions on Signal and Information Processing*, 3, 2014.

[33] Petros Drineas and Michael W Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *journal of machine learning research*, 6(Dec):2153–2175, 2005.

[34] Michael B Eisenberg. Information literacy: Essential skills for the information age. *DESIDOC journal of library & information technology*, 28(2):39, 2008.

[35] Scott Elias. *Implications of online social network sites on the personal and professional learning of educational leaders*. PhD thesis, Colorado State University, 2012.

[36] P Erd6s and A Renyi. On random graphs. *Publ. Math. Debrecen*, 6:290–297, 1959.

[37] Lifestyle Finder. Intelligent user profiling using large-scale demographic data. *Artificial Intelligence Magazine. v18 i2*, pages 37–45.

[38] Tom Forester. *The information technology revolution*. MIT Press, 1985.

[39] Alison Fox and Terese Bird. The challenge to professionals of using social media: Teachers in england negotiating personal-professional identities. *Education and Information Technologies*, 22(2):647–675, 2017.

[40] Ryan Glover. Building a twitter clone. *The Meteor Chef*, August 2017. [Online; posted 31-August-2017].

[41] Oana Goga, Howard Lei, Sree Hari Krishnan Parthasarathi, Gerald Friedland, Robin Sommer, and Renata Teixeira. Exploiting innocuous activity for correlating users across sites. In *Proceedings of the 22nd international conference on World Wide Web*, pages 447–458. ACM, 2013.

[42] Oana Goga, Daniele Perito, Howard Lei, Renata Teixeira, and Robin Sommer. Large-scale correlation of accounts across social networks. *University of California at Berkeley, Berkeley, California, Tech. Rep. TR-13-002*, 2013.

[43] Oana Goga, Giridhari Venkatadri, and Krishna P Gummadi. The doppelgänger bot attack: Exploring identity impersonation in online social networks. In *Proceedings of the 2015 Internet Measurement Conference*, pages 141–153. ACM, 2015.

[44] Sandra González-Bailón, Ning Wang, Alejandro Rivero, Javier Borge-Holthoefer, and Yamir Moreno. Assessing the bias in samples of large online networks. *Social Networks*, 38:16–27, 2014.

[45] Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018.

[46] Jeremy Greenwood and Boyan Jovanovic. The information-technology revolution and the stock market. *American Economic Review*, 89(2):116–122, 1999.

[47] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

[48] The Guardian. The guardian, March 2018. [Online; posted 17-March-2018].

[49] Aditi Gupta and Rishabh Kaushal. Towards detecting fake user accounts in facebook. In *Asia Security and Privacy (ISEASP), 2017 ISEA*, pages 1–6. IEEE, 2017.

[50] Asmelash Teka Hadgu and Jayanth Kumar Reddy Gundam. Learn2link: Linking the social and academic profiles of researchers. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 240–249, 2020.

[51] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[52] Sara Hajian, Francesco Bonchi, and Carlos Castillo. Algorithmic bias: From discrimination discovery to fairness-aware data mining. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 2125–2126. ACM, 2016.

[53] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[54] Joshua Hardwick. Google search operators: The complete list (42 advanced operators), December 2019. [Online; posted 24-December-2019].

[55] Bing-Zhe He, Chien-Ming Chen, Yi-Ping Su, and Hung-Min Sun. A defence scheme against identity theft attack based on multiple social networks. *Expert Systems with Applications*, 41(5):2345–2352, 2014.

[56] Mark Heimann, Haoming Shen, Tara Safavi, and Danai Koutra. Regal: Representation learning-based graph alignment. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 117–126. ACM, 2018.

[57] Jeff Hemsley. Social media giants are restricting research vital to journalism, July 2019. [Online; posted 11-July-2019].

[58] Luis Herranz, Shuqiang Jiang, and Xiangyang Li. Scene recognition with cnns: objects, scales and dataset bias. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 571–579, 2016.

[59] Susan C Herring and Sanja Kapidzic. Teens, gender, and self-presentation in social media. *International encyclopedia of social and behavioral sciences*, 2:1–16, 2015.

[60] Robert A Hinde. *Biological bases of human social behaviour*. McGraw-Hill, 1974.

[61] John R Hollenbeck and Bradley B Jamieson. Human capital, social capital, and social network analysis: Implications for strategic human resource management. *Academy of Management Perspectives*, 29(3):370–385, 2015.

[62] Markus Huber, Martin Mulazzani, Manuel Leithner, Sebastian Schrittwieser, Gilbert Wondracek, and Edgar Weippl. Social snapshots: Digital forensics for online social networks. In *Proceedings of the 27th annual computer security applications conference*, pages 113–122, 2011.

[63] Tereza Iofciu, Peter Fankhauser, Fabian Abel, and Kerstin Bischoff. Identifying users across social tagging systems. In *Fifth International AAAI Conference on Weblogs and Social Media*, 2011.

[64] Mansoor Iqbal. Youtube revenue and usage statistics (2019), August 2019. [Online; posted 8-August-2019].

[65] Paridhi Jain. Automated methods for identity resolution across heterogeneous social platforms. In *Proceedings of the 26th ACM Conference on Hypertext & Social Media*, pages 307–310, 2015.

[66] Paridhi Jain and Ponnurangam Kumaraguru. On the dynamics of username changing behavior on twitter. In *Proceedings of the 3rd IKDD Conference on Data Science, 2016*, pages 1–6, 2016.

[67] Paridhi Jain, Ponnurangam Kumaraguru, and Anupam Joshi. @ i seek'fb. me': Identifying users across multiple online social networks. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1259–1268. ACM, 2013.

[68] Paridhi Jain, Tiago Rodrigues, Gabriel Magno, Ponnurangam Kumaraguru, and Virgílio Almeida. Cross-pollination of information in online social media: A case study on popular social networks. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, pages 477–482. IEEE, 2011.

[69] J.Clement. Number of monthly active users in twitter, August 2019. [Online; posted 14-August-2019].

[70] Hao Jiang and John M Carroll. Social capital, social network and identity bonds: a reconceptualization. In *Proceedings of the fourth international conference on Communities and technologies*, pages 51–60, 2009.

[71] Lei Jin, Hassan Takabi, and James BD Joshi. Towards active detection of identity clone attacks on online social networks. In *Proceedings of the first ACM conference on Data and application security and privacy*, pages 27–38. ACM, 2011.

[72] Rishabh Kaushal, Srishti Chandok, Paridhi Jain, Prateek Dewan, Nalin Gupta, and Ponnurangam Kumaraguru. Nudging nemo: Helping users control linkability across social networks. In *International Conference on Social Informatics*, pages 477–490. Springer, 2017.

[73] Rishabh Kaushal, Vasundhara Ghose, and Ponnurangam Kumaraguru. Methods for user profiling across social networks. In *Proceedings of the 12th IEEE International Conference On Social Computing (SocialCom)*, pages 2104–2111. IEEE, 2019.

[74] Rishabh Kaushal, Shubham Gupta, and Ponnurangam Kumaraguru. Investigation of biases in identity linkage datasets. In *Proceedings of the 35th ACM/SIGAPP Symposium on Applied Computing*, pages 2104–2111. ACM, 2020.

[75] Rishabh Kaushal, Shubham Singh, and Ponnurangam Kumaraguru. Nexlink: Node embedding framework for cross-network linkages across social networks. In *Proceedings of the International Conference On Network Science (NetSciX)*, pages 2104–2111. Springer, 2020.

[76] Morteza Yousefi Kharaji, Fatemeh Salehi Rizi, and Mohammad Reza Khayyambashi. A new approach for finding cloned profiles in online social networks. *arXiv preprint arXiv:1406.7377*, 2014.

[77] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision*, pages 158–171. Springer, 2012.

[78] Xiangnan Kong, Jiawei Zhang, and Philip S Yu. Inferring anchor links across multiple heterogeneous social networks. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 179–188. ACM, 2013.

[79] Xiangnan Kong, Jiawei Zhang, and Philip S Yu. Inferring anchor links across multiple heterogeneous social networks. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 179–188. ACM, 2013.

[80] Nitish Korula and Silvio Lattanzi. An efficient reconciliation algorithm for social networks. *Proceedings of the VLDB Endowment*, 7(5):377–388, 2014.

[81] Ponnurangam Kumaraguru and Lorrie Faith Cranor. *Privacy indexes: a survey of Westin's studies*. Carnegie Mellon University, School of Computer Science, Institute for ..., 2005.

[82] Ronald La Due Lake and Robert Huckfeldt. Social capital, social networks, and political participation. *Political Psychology*, 19(3):567–584, 1998.

[83] Annabel Latham. Scroll.in, March 2018. [Online; posted 22-March-2018].

[84] Ronald Leenes. Context is everything sociality and privacy in online social network sites. In *IFIP PrimeLife International Summer School on Privacy and Identity Management for Life*, pages 48–65. Springer, 2009.

[85] Maurizio Lenzerini. Data integration: A theoretical perspective. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 233–246. ACM, 2002.

[86] Bruno Lepri, Nuria Oliver, Emmanuel Letouzé, Alex Pentland, and Patrick Vinck. Fair, transparent, and accountable algorithmic decision-making processes. *Philosophy & Technology*, 31(4):611–627, 2018.

[87] Jiexun Li, G Alan Wang, and Hsinchun Chen. Identity matching using personal and social identity features. *Information Systems Frontiers*, 13(1):101–113, 2011.

[88] Yongjun Li, You Peng, Zhen Zhang, Quanqing Xu, and Hongzhi Yin. Understanding the user display names across social networks. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 1319–1326. International World Wide Web Conferences Steering Committee, 2017.

[89] Shangsong Liang, Xiangliang Zhang, Zhaochun Ren, and Evangelos Kanoulas. Dynamic embeddings for user profiling in twitter. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1764–1773. ACM, 2018.

[90] Bang Hui Lim, Dongyuan Lu, Tao Chen, and Min-Yen Kan. # mytweet via instagram: Exploring user behaviour across multiple social networks. In *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 113–120. IEEE, 2015.

[91] Allison Lips. Everyone wants to get verified on social media, but it's not usually an easy process. *Social Media Week*, March 2018. [Online; posted 16-March-2018].

[92] Jie Liu, Zhicheng He, Lai Wei, and Yalou Huang. Content to node: Self-translation network embedding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1794–1802. ACM, 2018.

[93] Jing Liu, Fan Zhang, Xinying Song, Young-In Song, Chin-Yew Lin, and Hsiao-Wuen Hon. What's in a name?: an unsupervised approach to link users across communities. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 495–504. ACM, 2013.

[94] Li Liu, William K Cheung, Xin Li, and Lejian Liao. Aligning users across social networks using network embedding. In *IJCAI*, pages 1774–1780, 2016.

[95] Siyuan Liu, Shuhui Wang, Feida Zhu, Jinbo Zhang, and Ramayya Krishnan. Hydra: Large-scale social identity linkage via heterogeneous behavior modeling. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 51–62. ACM, 2014.

[96] Yufei Liu, Dechang Pi, and Lin Cui. Learning user distance from multiple social networks. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 3280–3287. IEEE, 2017.

[97] George Loewenstein, Cass R Sunstein, and Russell Golman. Disclosure: Psychology changes everything. *Annu. Rev. Econ.*, 6(1):391–419, 2014.

[98] David Loshin. *The practitioner's guide to data quality improvement.* Elsevier, 2010.

[99] Binh Thanh Luong, Salvatore Ruggieri, and Franco Turini. k-nn as an implementation of situation testing for discrimination discovery and prevention. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 502–510. ACM, 2011.

[100] Suman Kalyan Maity, Aishik Chakraborty, Pawan Goyal, and Animesh Mukherjee. Detection of sockpuppets in social media. In *Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 243–246, 2017.

[101] Anshu Malhotra, Luam Totti, Wagner Meira Jr, Ponnurangam Kumaraguru, and Virgilio Almeida. Studying user footprints in different online social networks. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, pages 1065–1070. IEEE Computer Society, 2012.

[102] Tong Man, Huawei Shen, Shenghua Liu, Xiaolong Jin, and Xueqi Cheng. Predict anchor links across social networks via an embedding approach. In *IJCAI*, volume 16, pages 1823–1829, 2016.

[103] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge university press, 2008.

[104] Steve Mansfield-Devine. Google hacking 101. *Network Security*, 2009(3):4–6, 2009.

[105] Louise Matsakis. The wired guide to your personal data (and who is using it), February 2019. [Online; posted 15-February-2019].

[106] Ryuta Matsuno and Tsuyoshi Murata. Mell: effective embedding method for multiplex networks. In *Companion Proceedings of the The Web Conference 2018*, pages 1261–1268, 2018.

[107] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.

[108] Front Gate Media. List of 723 bad words to blacklist & how to use facebook's moderation tool. *Front Gate Media*, May 2014. [Online; posted 12-May-2014].

[109] Mary C Michel, Marco Carvalho, Heather Crawford, and Albert C Esterline. Cyber identity: Salient trait ontology and computational framework to aid in solving cybercrime. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 1242–1249. IEEE, 2018.

[110] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[111] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[112] Xin Mu, Feida Zhu, Ee-Peng Lim, Jing Xiao, Jianzong Wang, and Zhi-Hua Zhou. User identity linkage by latent user space modelling. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1775–1784, 2016.

[113] San Murugesan. Understanding web 2.0. *IT professional*, 9(4):34–41, 2007.

[114] Cataldo Musto, Pierpaolo Basile, Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Linked open data-enabled strategies for top-n recommendations. In *CBRecSys@ RecSys*, pages 49–56. Citeseer, 2014.

[115] Yaroslav Nechaev, Francesco Corcoglioniti, and Claudio Giuliano. Sociallink: exploiting graph embeddings to link dbpedia entities to twitter profiles. *Progress in Artificial Intelligence*, 7(4):251–272, 2018.

[116] Yuanping Nie, Yan Jia, Shudong Li, Xiang Zhu, Aiping Li, and Bin Zhou. Identifying users across social networks based on dynamic core interests. *Neurocomputing*, 210:107–115, 2016.

[117] Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Emre Kiciman. Social data: Biases, methodological pitfalls, and ethical boundaries. *Frontiers in Big Data*, 2:13, 2019.

[118] Vito Claudio Ostuni, Tommaso Di Noia, Eugenio Di Sciascio, and Roberto Mirizzi. Top-n recommendations from implicit feedback leveraging linked open data. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 85–92, 2013.

[119] Makbule Gulcin Ozsoy, Faruk Polat, and Reda Alhajj. Making recommendations by integrating information from multiple social networks. *Applied Intelligence*, 45(4):1047–1065, 2016.

[120] Rachel Palmateer. Key linkedin statistics to know from 2019, October 2019. [Online; posted 19-October-2019].

[121] Bimal Parmar. Protecting against spear-phishing. *Computer Fraud & Security*, 2012(1):8–11, 2012.

[122] Daniele Perito, Claude Castelluccia, Mohamed Ali Kaafar, and Pere Manils. How unique and traceable are usernames? In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 1–17. Springer, 2011.

[123] Daniele Perito, Claude Castelluccia, Mohamed Ali Kaafar, and Pere Manils. How unique and traceable are usernames? In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 1–17. Springer, 2011.

[124] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

[125] Evaggelia Pitoura, Panayiotis Tsaparas, Giorgos Flouris, Irini Fundulaki, Panagiotis Papadakos, Serge Abiteboul, and Gerhard Weikum. On measuring bias in online information. *ACM SIGMOD Record*, 46(4):16–21, 2018.

[126] Iasonas Polakis, Georgios Kontaxis, Spiros Antonatos, Eleni Gessiou, Thanasis Petsas, and Evangelos P Markatos. Using social networks to harvest email addresses. In *Proceedings of the 9th Annual ACM Workshop on Privacy in the Electronic Society*, pages 11–20, 2010.

[127] Facebook Inc. Press. Company info: Facebook, February 2019. [Online; posted 11-February-2019].

[128] Guo-Jun Qi, Charu C Aggarwal, and Thomas Huang. Link prediction across networks by biased cross-network sampling. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pages 793–804. IEEE, 2013.

[129] Maria Ranieri, Stefania Manca, and Antonio Fini. Why (and how) do teachers engage in social networks? an exploratory study of professional use of f acebook and its implications for lifelong learning. *British journal of educational technology*, 43(5):754–769, 2012.

[130] Christopher Riederer, Yunsung Kim, Augustin Chaintreau, Nitish Korula, and Silvio Lattanzi. Linking users across domains with location data: Theory and validation. In *Proceedings of the 25th International Conference on World Wide Web*, pages 707–719. International World Wide Web Conferences Steering Committee, 2016.

[131] Andrea Romei and Salvatore Ruggieri. A multidisciplinary survey on discrimination analysis. *The Knowledge Engineering Review*, 29(5):582–638, 2014.

[132] Grant James Ryan, Shaun William Ryan, Craig Matthew Ryan, Wayne Alistar Munro, and Del Robinson. Search engine, July 16 2002. US Patent 6,421,675.

[133] Sina Sajadmanesh, Hamid R Rabiee, and Ali Khodadadi. Predicting anchor links between heterogeneous social networks. In *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 158–163. IEEE Press, 2016.

[134] Monica Scannapieco, Paolo Missier, and Carlo Batini. Data quality at a glance. *Datenbank-Spektrum*, 14(January):6–14, 2005.

[135] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

[136] Florian Schaub, Rebecca Balebako, Adam L Durity, and Lorrie Faith Cranor. A design space for effective privacy notices. In *Eleventh Symposium On Usable Privacy and Security ({SOUPS} 2015)*, pages 1–17, 2015.

[137] Alex Scroxton. Computerweekly, April 2019. [Online; posted 25-April-2019].

[138] Indira Sen, Anupama Aggarwal, Shiven Mian, Siddharth Singh, Ponnurangam Kumaraguru, and Anwitaman Datta. Worth its weight in likes: Towards detecting fake likes on instagram. In *Proceedings of the 10th ACM Conference on Web Science*, pages 205–209. ACM, 2018.

[139] Andrew Perrin Shannon Greenwood and Maeve Duggan. Social media updates (2016), November 2016. [Online; posted November-2016].

[140] Yilin Shen and Hongxia Jin. Controllable information sharing for user accounts linkage across multiple online social networks. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 381–390. ACM, 2014.

[141] Yilin Shen and Hongxia Jin. Controllable information sharing for user accounts linkage across multiple online social networks. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 381–390. ACM, 2014.

[142] Meredith M Skeels and Jonathan Grudin. When social networks cross boundaries: a case study of workplace use of facebook and linkedin. In *Proceedings of the ACM 2009 international conference on Supporting group work*, pages 95–104, 2009.

[143] Hoax Slayer. What is facebook cloning and how can i protect myself from it? *Hoax Slayer*, July 2017. [Online; posted 25-July-2017].

[144] Jason Slotkin. Twitter 'bots' steal tweeters' identities. *Market Place*, May 2013. [Online; posted 27-May-2013].

[145] Charles Steinfield, Nicole B Ellison, and Cliff Lampe. Social capital, self-esteem, and use of online social network sites: A longitudinal analysis. *Journal of Applied Developmental Psychology*, 29(6):434–445, 2008.

[146] Sen Su, Li Sun, Zhongbao Zhang, Gen Li, and Jielun Qu. Master: across multiple social networks, integrate attribute and structure embedding for reconciliation. In *IJCAI*, pages 3863–3869, 2018.

[147] Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. Bootstrapping entity alignment with knowledge graph embedding. In *IJCAI*, pages 4396–4402, 2018.

[148] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. International World Wide Web Conferences Steering Committee, 2015.

[149] Flavio Toffalini, Maurizio Abbà, Damiano Carra, and Davide Balzarotti. Google dorks: Analysis, creation, and new defenses. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 255–275. Springer, 2016.

[150] Tatiana Tommasi, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars. A deeper look at dataset bias. In *Domain adaptation in computer vision applications*, pages 37–55. Springer, 2017.

[151] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011.

[152] Jonathan H Turner. *Human institutions: A theory of societal evolution*. Rowman & Littlefield, 2003.

[153] José Van Dijck. 'you have one identity': performing the self on facebook and linkedin. *Media, culture & society*, 35(2):199–215, 2013.

[154] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P Gummadi. On the evolution of user interaction in facebook. In *Proceedings of the 2nd ACM workshop on Online social networks*, pages 37–42. ACM, 2009.

[155] Bimal Viswanath, Mainack Mondal, Allen Clement, Peter Druschel, Krishna P Gummadi, Alan Mislove, and Ansley Post. Exploring the design space of social network-based sybil defenses. In *Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on*, pages 1–8. IEEE, 2012.

[156] Jessica Vitak. The impact of context collapse and privacy on social network site disclosures. *Journal of broadcasting & electronic media*, 56(4):451–470, 2012.

[157] Benjamin W Wah. Generalization and generalizability measures. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):175–186, 1999.

[158] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1225–1234. ACM, 2016.

[159] Gang Wang, Manish Mohanlal, Christo Wilson, Xiao Wang, Miriam Metzger, Haitao Zheng, and Ben Y Zhao. Social turing tests: Crowdsourcing sybil detection. *arXiv preprint arXiv:1205.3856*, 2012.

[160] Yang Wang, Pedro Giovanni Leon, Alessandro Acquisti, Lorrie Faith Cranor, Alain Forget, and Norman Sadeh. A field trial of privacy nudges for facebook. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 2367–2376, 2014.

[161] Yang Wang, Pedro Giovanni Leon, Xiaoxuan Chen, and Saranga Komanduri. From facebook regrets to facebook privacy nudges. *Ohio St. LJ*, 74:1307, 2013.

[162] Yang Wang, Pedro Giovanni Leon, Kevin Scott, Xiaoxuan Chen, Alessandro Acquisti, and Lorrie Faith Cranor. Privacy nudges for social media: an exploratory facebook study. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 763–770, 2013.

[163] Yang Wang, Gregory Norcie, Saranga Komanduri, Alessandro Acquisti, Pedro Giovanni Leon, and Lorrie Faith Cranor. " i regretted the minute i pressed share" a qualitative study of regrets on facebook. In *Proceedings of the seventh symposium on usable privacy and security*, pages 1–16, 2011.

[164] Yaqing Wang, Chunyan Feng, Ling Chen, Hongzhi Yin, Caili Guo, and Yunfei Chu. User identity linkage across social networks via linked heterogeneous network embedding. *World Wide Web*, pages 1–22, 2018.

[165] Yubin Wang, Tingwen Liu, Qingfeng Tan, Jinqiao Shi, and Li Guo. Identifying users across different sites using usernames. *Procedia Computer Science*, 80:376–385, 2016.

[166] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440, 1998.

[167] Wei Xie, Xin Mu, Roy Ka-Wei Lee, Feida Zhu, and Ee-Peng Lim. Unsupervised user identity linkage via factoid embedding. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1338–1343. IEEE, 2018.

[168] Linchuan Xu, Xiaokai Wei, Jiannong Cao, and Philip S Yu. On exploring semantic meanings of links for embedding social networks. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 479–488. International World Wide Web Conferences Steering Committee, 2018.

[169] Jaewon Yang and Jure Leskovec. Patterns of temporal variation in online media. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 177–186. ACM, 2011.

[170] Wan-Shiou Yang, Jia-Ben Dia, Hung-Chi Cheng, and Hsing-Tzu Lin. Mining social networks for targeted advertising. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, volume 6, pages 137a–137a. IEEE, 2006.

[171] Yang Yang, De-Chuan Zhan, Yi-Feng Wu, and Yuan Jiang. Multi-network user identification via graph-aware embedding. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 209–221. Springer, 2018.

[172] Abdurrahman Yasar and Ümit V Çatalyürek. An iterative global structure-assisted labeled network aligner. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2614–2623, 2018.

[173] Haifeng Yu, Michael Kaminsky, Phillip B Gibbons, and Abraham Flaxman. Sybilguard: defending against sybil attacks via social networks. In *ACM SIGCOMM Computer Communication Review*, volume 36, pages 267–278. ACM, 2006.

[174] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. *arXiv preprint arXiv:1507.05259*, 2015.

[175] Reza Zafarani and Huan Liu. Connecting users across social media sites: a behavioral-modeling approach. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 41–49. ACM, 2013.

[176] Reza Zafarani and Huan Liu. Users joining multiple sites: Friendship and popularity variations across sites. *Information Fusion*, 28:83–89, 2016.

[177] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. Fa* ir: A fair top-k ranking algorithm. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1569–1578. ACM, 2017.

[178] Bo Zhang and Heng Xu. Privacy nudges for mobile applications: Effects on the creepiness emotion and privacy attitudes. In *Proceedings of the 19th ACM conference on computer-supported cooperative work & social computing*, pages 1676–1690, 2016.

[179] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. Network representation learning: A survey. *IEEE transactions on Big Data*, 2018.

[180] Haochen Zhang, Minyen Kan, Yiqun Liu, and Shaoping Ma. Online social network profile linkage based on cost-sensitive feature acquisition. In *Chinese National Conference on Social Media Processing*, pages 117–128. Springer, 2014.

[181] Jialong Zhang, Jayant Notani, and Guofei Gu. Characterizing google hacking: A first large-scale quantitative study. In *International Conference on Security and Privacy in Communication Networks*, pages 602–622. Springer, 2014.

[182] Jiawei Zhang, Xiangnan Kong, and Philip S Yu. Transferring heterogeneous links across location-based social networks. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 303–312. ACM, 2014.

[183] Jiawei Zhang and S Yu Philip. Link prediction across heterogeneous social networks: A survey. *Social networks*, 2014.

[184] Jiawei Zhang and S Yu Philip. Integrated anchor and social link predictions across social networks. In *IJCAI*, pages 2125–2132, 2015.

[185] Jiawei Zhang and S Yu Philip. Multiple anonymized social networks alignment. In *2015 IEEE International Conference on Data Mining*, pages 599–608. IEEE, 2015.

[186] Jiawei Zhang and Philip S Yu. Pct: partial co-alignment of social networks. In *Proceedings of the 25th International Conference on World Wide Web*, pages 749–759. International World Wide Web Conferences Steering Committee, 2016.

[187] Jiawei Zhang, Philip S Yu, and Zhi-Hua Zhou. Meta-path based multi-network collective link prediction. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1286–1295. ACM, 2014.

[188] Jing Zhang, Bo Chen, Xianming Wang, Hong Chen, Cuiping Li, Fengmei Jin, Guojie Song, and Yutao Zhang. Mego2vec: Embedding matched ego networks for user alignment across social networks. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 327–336, 2018.

[189] Yutao Zhang, Jie Tang, Zhilin Yang, Jian Pei, and Philip S Yu. Cosnet: Connecting heterogeneous social networks with local and global consistency. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1485–1494. ACM, 2015.

[190] Yuxiang Zhang, Jiamei Fu, Chengyi Yang, and Chunjing Xiao. A local expansion propagation algorithm for social link identification. *Knowledge and Information Systems*, 60(1):545–568, 2019.

[191] Zhongbao Zhang, Qihang Gu, Tong Yue, and Sen Su. Identifying the same person across two similar social networks in a unified way: Globally and locally. *Information Sciences*, 394:53–67, 2017.

[192] Fan Zhou, Lei Liu, Kunpeng Zhang, Goce Trajcevski, Jin Wu, and Ting Zhong. Deeplink: A deep learning approach for user identity linkage. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 1313–1321. IEEE, 2018.

[193] Jingya Zhou and Jianxi Fan. Translink: User identity linkage across heterogeneous social networks via translating embeddings. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 2116–2124. IEEE, 2019.

[194] Xiaoping Zhou, Xun Liang, Xiaoyong Du, and Jichao Zhao. Structure based user identification across social networks. *IEEE Transactions on Knowledge and Data Engineering*, 30(6):1178–1191, 2018.

[195] Xiaoping Zhou, Xun Liang, Haiyan Zhang, and Yuefeng Ma. Cross-platform identification of anonymous identical users in multiple social media networks. *IEEE transactions on knowledge and data engineering*, 28(2):411–424, 2016.

[196] Jan Henrik Ziegeldorf, Martin Henze, René Hummen, and Klaus Wehrle. Comparison-based privacy: nudging privacy in social media (position paper). In *Data Privacy Management, and Security Assurance*, pages 226–234. Springer, 2015.

[197] Indre Zliobaite. A survey on measuring indirect discrimination in machine learning. *arXiv preprint arXiv:1511.00148*, 2015.