A Differentially Private Matching Scheme for Pairing Similar Users of Proximity Based Social Networking applications

Michael Sommer, Lipyeow Lim, and Depeng Li
Department of Information and Computer Sciences
University of Hawaii at Manoa, Honolulu, HI, USA, 96822
{mpsommer, lipyeow, depengli}@hawaii.edu

Abstract

The pervasiveness of smartphones has made connecting with users through proximity based mobile social networks commonplace in today's culture. Many such networks connect users by matching them based on shared interests. With ever-increasing concern for privacy, users are wary of openly sharing personal information with strangers. Several methods have addressed this privacy concern such as encryption and k-anonymity, but none address issues of eliminating third party matches, achieving relevant matches, and prohibiting malicious users from inferring information based on their input into the system. In this paper, we propose a matching scheme that accurately pairs similar users while simultaneously providing protection from malicious users inferring information. Specifically, we match users in a proximity-based social network setting adapted from a framework of differential privacy. This eliminates the need for third-party matching schemes, allows for accurate matching, and ensures malicious users will be unable to infer information from matching results.

1. Introduction

The reliance on mobile devices is ever increasing in our current culture. The pervasiveness of such devices allows for such conveniences as Location Based Services (LBSs). LBSs are mobile application services that provide a service, such as a restaurant recommendation, to a user based on their location. Foursquare and Yelp are two widely known LBSs. Social networking applications often utilize LBSs in Proximity Based Social Networking (PBSN), which is commonly used to match users based on similarities in interests in things like music, sports, movies, etc. Examples of PBSN's are mobile dating applications such as Tinder, and Sonar.me. A typical process for matching users in a PBSN is for an initiating party to broadcast a user's interests to the other nearby users directly. These users then decide if a connection will be made based on similarities between them and the initiator. The result will either be a match, or no match, based on some similarity computation. Although this scheme is effective, unfortunately, it sacrifices the privacy of the users. Users might not feel comfortable broadcasting their interests if it is related to sensitive matters. It is not hard to imagine a scenario where a user maintains a personal persona separate from their professional persona. Bad things could happen if this user were matched with a fellow employee who might reveal sensitive information about the user in the workplace. Also, users may experience loss of privacy if a malicious user systematically changes their interests until a match is made with another user. It allows the malicious user to infer information about other users interests by simply using the application how it is intended. Common approaches of encryption and k-anonymity do not successfully address preserving user privacy while simultaneously producing accurate results.

1.1. Related Word

Various solutions have been proposed in recent years [1,7,8,9,13,14] to address the issues of preserving privacy while matching users. These solutions all assume the user has multiple interests chosen from a public set of defined interests. These interests range from things like what type of music a person likes, to how often do they consume alcohol.

[10] matches users by both the number of common interests and the corresponding interest weight on each of the individual users. They argue this allows for more accurate fine-grained matches. Instead of peer to peer sharing of data to find user matches, [10] relies on a third-party matching scheme. The problem with this reliance on a third-party is that they maintain a central repository for all user data and therefore are subject to a central point of failure if the third-party entity is

URI: http://hdl.handle.net/10125/50118
ISBN: 978-0-9981331-1-9

somehow compromised. [10] also relies on computationally expensive encryption techniques that do not address the issue of a malicious user inferring other user information based on their interests and match results.

[6] incorporates differential privacy by perturbing data with noise from the Laplace distribution and using secure multi-party computation (SMC) for matching. They utilize a blocking step when processing the data for SMC. This filters out records that will not be part of the join result and ensures the cost of matching during SMC will be at acceptable levels. Instead of using common k-anonymity techniques to sanitize the data, they leverage differential privacy. Essentially, they group records according to their attributes and then use noise drawn from the Laplacian distribution to either suppress a record or add a fake record. If the noise is positive they add fake records to the dataset, and if it is negative they suppress records from the dataset. Their experimental analysis aims at reducing the cost of private matching records. Although they show their approach provides strong privacy guarantees, their effectiveness in reducing matching costs is approximately the same as the k-anonymity versions.

The approach in [6] essentially uses the noise from the Laplacian distribution as a binary decision to add or suppress data. This is fundamentally different from our approach. We add noise from the Laplace distribution to all attributes of a given record. The experimental results of [6] also focus on reducing the cost of SMC so it is unclear as to how accurate their matching results are.

[7] develops an approach to publish search queries and click data in the form of a click query graph. A query is only published if the query frequency plus some added noise exceeds some threshold. Their findings are that the more stringent the privacy requirement, the higher the threshold, and consequently the fewer the number of queries that can be safely published. Their approach is similar to [6] in the way they information is released based on some threshold. Our research differs in the way we use Laplacian noise with data. We don not suppress data based on some threshold as [7] does. It is also unclear if the approach in [7] relies on third parties. The experimental analysis of [7] found that keywords obtained with the perturbed data closely resemble the original unperturbed data. Whether this translates to a more general case of matching users is unclear.

[8] avoids the issues with third party matching by directly calculating matches between two users based on a maximal intersection of interest sets where interest is defined as a string up to a certain length. Calculations are computed locally on each user's machine. It is unclear how accurate the matches between users is with this approach, and no data is provided regarding accuracy. Also, the set matching technique in [8] does not address the issue of malicious users inferring information because when one user attempts to match with another, they can infer interests of the other from the matching results.

[13] uses three different protocols to match users in PBSN's. These protocols assume an honest but curious user and do not address the scenario of a malicious user who changes input and observes output to infer information of another user. The paper states that if a malicious user Bob repeatedly tries to match with a user Alice by creating fake profiles, he can eventually learn Alice's profile information. To combat this problem, they suggest limiting the number of times that Bob can attempt to match to Alice. Unfortunately, this suggestion is not an adequate solution to the problem our paper focuses on; ensuring that user profile information is safe from malicious users. Although [13] provides adequate protocols for an honest but curious scenario, they do not address the problems that are the focus of our paper.

[14] matches users with similarity calculations based on prioritizing and weighting individual interests. This is done without the use of a third party matching scheme, but a malicious user can still infer knowledge by repeatedly querying nearby users and analyzing results. Again the accuracy of matches is not thoroughly examined in this research.

The problem of secure matching discussed in this paper is similar to secure recommender systems. Cryptographic solutions to the problem of secure recommender systems such as [5,6] focus on removing the third-party recommenders. They do not attempt to limit the amount of knowledge malicious users can infer through using the system as intended.

[9] develops a differentially private recommender system that provides guarantees of privacy for users. Their system ensures some level of privacy against malicious users inferring information from the use of systems that recommend movies from the Netflix dataset. The approach in this work is developed for a specific recommender system that uses certain statistical steps. This approach does not generalize to all recommenders and therefore does not work for our research. Technical differences between our work are in the way we use the interest weight of a user as the count query function where they compute a covariance matrix. Lastly, the general objective of the research in [9] is different from ours. [9] uses statistical methods to aggregate user data and recommend movies where our work involves mutual selection between two parties by focusing on user to user matching, not an

aggregation of some number of user records to recommend something to a given user.

[11] develops a lightweight recommender system that uses perturbed data for existing recommender systems. Data is perturbed locally and their system works with existing recommenders. Perturbing the data locally ensures their objective of privacy preservation is met and allows their data to be used with third party recommenders. Their experimental analysis shows that recommendation accuracy is preserved while perturbing the data. Our research differs in the general objective. Similar to the differences with [9], [11] focuses on recommending products to a user based on aggregating some number of user records to find the most relevant product. It is unclear how this approach would generalize to our objective of finding the best match from direct user to user matching.

A common approach to preserving privacy in data is anonymization. This involves methods such as k-anonymity. K-anonymity typically attempts to anonymize data with two techniques: suppression and generalization. Suppression involves removing or masking certain attributes in the dataset and generalization involves replacing individual values of attributes with a general range. The problem with these approaches as shown in [1] is that the anonymization techniques often used render the data useless for matching algorithms. Differential privacy does not experience this problem.

1.1. Contributions

The contributions of this work are to implement a matching algorithm to be used for PBSN's that provides differential privacy guarantees for the users. The challenge of this task is to provide accurate matching while maintaining privacy. This challenge is overcome by developing a matching algorithm within the framework of differential privacy. The issue of using a third-party matching scheme is arbitrarily solved by calculating the matching metrics for users directly on their devices. Furthermore, third-party matching schemes could be used as long as the data they received was already differentially private. The issue of malicious users inferring user information from the results of the algorithm is addressed by applying differentially private techniques to user's interest data. Specifically, perturbing the data using random noise drawn from the Laplace distribution. Applying the Laplace mechanism to each user's interest data set ensures that a malicious user will not be able to determine a user's interests and that subsequent queries will also not leak information. Users privacy will be guaranteed up to some ϵ, meaning no user will be able to definitively infer each interest level of another user. The similarity between two users is calculated with the Pearson coefficient similarity metric, a common similarity metric used in matching algorithms.

2. Preliminaries

2.1. Dataset and Distance

In this research, a dataset x is a collection of interests from the universe U of all possible interests. When a given user does not have an interest in a particular attribute, the interest will remain in the dataset with a weight of 0. Our universe U of interests is U = {$interest_i$, …$interest_n$} where i = 1 … n and n is the total number of possible interests. Given this definition, we can define the distance between two datasets x, y with the l_1 norm as [3]:

$$\sum_{i=1}^{|U|} |x_1 - y_1|$$

The dataset used for our experiments consists of weight vectors of user interests. Weights for interests range from 0-5 where 0 indicates no preference (no interest) and 5 indicates strong preference. An attribute in the dataset is an individual weight for a user interest.

2.2. Differential Privacy

Differential privacy formalizes the idea that the output of some computation does not allow inference to be made on the presence or absence of any record in the computations input. More formally, it requires that for any outcome of a randomized computation, that outcome should be nearly equally likely with and without any one record [9]. A randomized computation M satisfies ϵ-differential privacy if for any adjacent dataset X and Y, and any subset S of possible outcomes Range(M),

$$\frac{\Pr[M(X) \in S]}{\Pr[M(X) \in S]} \leq e^{\epsilon}$$

The guarantee differential privacy provides can be interpreted as a bound on the ability to infer from any output event S, whether the input to the computation was X or Y [3]. In our project, this means inference about the presence or absence of any given attribute (user interest) is bounded by a factor of e^{ϵ}

It is important to note that differential privacy is a property used in our algorithm that outputs matches. It is not the output itself. Differential privacy is a privacy guarantee for some defined ϵ.

2.3. Laplace Mechanism

The Laplace Mechanism is used to ensure differential privacy with our matches. This works by perturbing a counting query f with noise distributed according to a Laplace distribution centered at 0 with scale $\beta = \frac{1}{\epsilon}$,

$$\text{Lap}(x|\beta) = \frac{1}{2\beta} e^{-\frac{|x|}{\beta}}$$

then the Laplace mechanism is defined as: $M_l(x, f, \epsilon) = f(x) + Z$ where Z is a random variable drawn from the Laplace distribution. The Laplace Mechanism is proven to ensure differential privacy [3]. In our work, the counting query function f is simply the preference count for a given interest. Similar to counting a population of people represented in a dataset that smoke, the preference count represents the number of bits that represent the amount of interest a user has in a specific area.

2.4. Similarity Metrics and Matching

Matching in our work is where for a given user Alice from a set of N users, a similarity metric is calculated for the N-1 users and the user with the highest similarity score to Alice is the one she is matched with. Each user has a set of interests represented as a weight vector and inputted into the Lapalce mechanism. The output of the Laplace mechanism from one user to the rest is used to compute the Pearson correlation coefficient (PCC), which computes the similarity between two users. PCC in this instance shows the linear relationship between two weight vectors. PCC output ranges between -1 and 1 where a value greater than 1 would be a positive relationship for the vectors, a value less than 0 would be a negative relationship, and a value of 0 would be no relationship at all.

2.5. Adversary Models

A common user scheme for this type of research is to consider the user honest-but-curious [10][3][8]. This research addresses the honest-but-curious user approach as well as the malicious user approach. This malicious user has the power to modify interest weight input until a match is made with a targeted user, or information is inferred about the user. In this scheme, any given user can act as the malicious user.

3. Problem Description

The protocol in our approach is a PBSN that involves n users geographically close, who are trying to match with each other based on shared interests. This protocol does not rely on interacting with a third party matching entity. The process is divided into two phases: creating differentially private user datasets, and matching users. An example assumes n users where each user has their own preference dataset $I_p = <I_{p1}, I_{p2}. \ldots, I_{pn}>$. The PBSN application ensures each users dataset is differentially private. Each user dataset is then shared with all n-1 users and the similarity is computed. The most similar user to a given user Alice is the one matched to Alice. This can also be generalized to the most similar k users, where k can equal 1 to n-1.

3.1. Phase I

In this phase, the Laplace mechanism is applied to a given user's preference dataset. This is done locally on each user's machine. This eliminates the need for a third-party matching system as well as sharing unperturbed data with other users. The Laplace mechanism adds random noise to each interest count drawn from the Laplace distribution. Once the Laplace mechanism outputs the perturbed dataset, it can be shared with all other n-1 users.

3.2. Phase II

In this phase, a user calculates their matching similarity with all n-1 users. This is done by calculating the PCC for a given user amongst all other users with their provided perturbed dataset. The user with the highest similarity to a user Alice will be the one she is matched with.

Since the datasets used in the matching satisfy differential privacy for some ϵ, Alice cannot determine definitively the exact interests and weights that achieved this match. Furthermore, each user can control the privacy they are comfortable with by adjusting the noise metric used in their differential privacy calculation.

4. Experiments

The experiments in this paper consist of simulations of PBSNs that pair similar users. The simulations are implemented in Python and the dataset used is a real world dataset pertaining to the interests, habits fears, and opinions of young people between the ages of 15 and 30 in Slovakia. The responses to each question are in the form of a scale where 0 denotes no interest and 5 denotes a strong interest [16]. There are a total of 1010 participants and a total of 140 attributes for each respondent. Three experiments are performed to analyze the trade-off in accuracy vs privacy, as well as validate accuracy and usefulness of our system. Experiment 1 calculates the number of features vs the matching accuracy for different amounts of noise (ϵ) drawn from the Laplace distribution. Experiment 2 computes the best match for one sample with an ϵ of 0. The placement of this best match in similarity lists (list of highest scoring matches to lowest scoring, for a given amount of noise) of various ϵ is compared. Experiment 3 computes the match for multiple values of ϵ. The rank of these matches on a baseline similarity list is compared.

4.1. Experiment 1

Experiment 1 compares the trade-off between accuracy and privacy (ϵ) with various feature sizes. Here, features represent the weight vectors of users interests. For each ϵ, we compute the best match with 10, 50, 90, and 130 features. A random sample is drawn from the dataset to compute the PCC against all other samples. The values for ϵ that we compare are 0.0, 0.1, 0.3, 0.5, 0.7, and 0.9. The baseline ϵ that we compare all others to is $\epsilon = 0$ which means no noise is added to the data. For the baseline, as the number of features included in the similarity calculation increases, the similarity score decreases. After computing the baseline for the various feature sizes, we then generate a similarity score between all users, for all values of ϵ. For each ϵ, we now have a list of similarity scores between all the samples and the randomly chosen one. From here, we use the highest scoring calculation from each feature size in the baseline and find it in the similarity lists for each ϵ. This allows us to compare the accuracy of the baseline match and the different amount of privacy applied to the system. The results of this are displayed in Figure 1.

4.2. Experiment 2

In this experiment, we compute the PCC for one randomly chosen sample against all other samples in our dataset. Our baseline used for comparison is when $\epsilon = 0$. Like before, we obtain lists of similarity scores by computing PCC between the random sample and all others for ϵ = 0.1, 0.3, 0.5, 0.7, 0.9. We do this for feature sizes of 10 to 140, where the feature size increases by 10 each time. This results in a total of 14 features sizes. We next compare the highest scoring match in the baseline at each feature size to the similarity lists for each ϵ size. Specifically, we calculate where the baseline match sits in the similarity list for each ϵ, and we store this in a matrix. From here, we can compute the average distance of how far away a value for ϵ places the baseline match across all feature sizes. The results of this are displayed in figure 2.

4.3. Experiment 3

Experiment 3 computes the PCC for one randomly chosen sample against all other samples in our dataset. Our baseline used for comparison is once again when $\epsilon = 0$. In this experiment we compute a similarity list for the baseline. We then compute the highest scoring match for each value of ϵ across all feature sizes (feature sizes are the same as experiment 2). After this, we check to see where the highest scoring match for each value of ϵ rank in the baseline similarity list. We essentially reverse the comparison done in experiment 2. Results of experiment 3 are found in figure 3.

5. Experimental Evaluation

The general findings of our experiments show that although there is a trade-off between accuracy and privacy when matching users, our system matches users with high similarity even with significant noise added to the data

5.1. Experiment 1 Evaluation

The results of experiment 1 reveal multiple findings. In general, figure 1 shows that as a number of features used in calculating similarity increases, the similarity scores decrease. It is obvious that when the feature set is more robust there will be less similarity between users, because there are more features to consider, resulting in a decrease in score. This holds true whether or not noise is added to the data.

The most important finding from experiment 1 is that as the noise added to the data increases, the

accuracy of the matching algorithm decreases, but not significantly for $\epsilon \leq 0.5$. This has been noted in previous research [9] and is now displayed in our novel approach. Our system maintains relatively high matching accuracy even with significant noise added to the data

The takeaway from this experiment is that PBSN's are capable of overcoming the challenge of not allowing malicious users to infer information thus preserving user privacy in matching algorithms. Figure 1 shows that depending on the data for a particular PBSN; consideration is required to find the right balance between a number of features used for calculating similarity, the amount of noise to add to the data, and how much variation in accuracy to allow in the system.

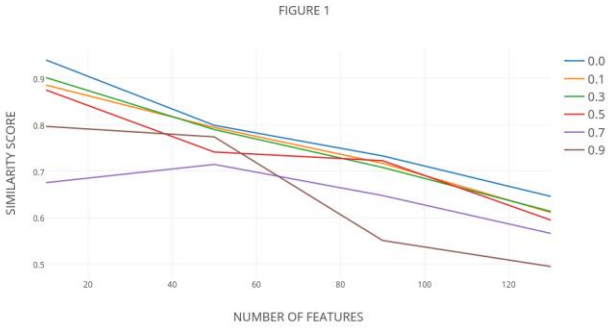

Fig. 1. **Each line represents some ϵ of privacy. The comparison of privacy vs accuracy is compared to various sizes of features used in computing similarity between two samples. The red line has no noise added, and serves as the comparison for all others.**

5.2. Experiment 2 Evaluation

Results from experiment 1 show that striking the right balance between feature size and privacy can allow our system to still perform highly accurate matching. Experiment 2 looks at the accuracy from a different perspective and aids in validating the findings from Experiment 1.

Results of experiment 2 show that for $\epsilon \leq 0.5$, the baseline highest scoring match for a given feature size is less than six users away from the top of the similarity lists for each ϵ on average. This is a significant finding because it shows that adding noise up to 0.5 keeps the relative order for the k closest matches intact and still matches users with high similarity. This means that although our system might not match the true highest scoring match, it matches on one of the top six. Even on our relatively small dataset of approximately 1000 users, the top ten closest matches were always within a similarity score with a range of no more than 10%.

The impact of this finding is great for PBSN's. It ensures that even if a differentially private system does not match a user to the user with the highest similarity score, it will be one of the top six. When more features are used to calculate the similarity between two users, similarity scores decrease. This means that the reality of a perfect match is more difficult to achieve. Because of this, our system works as well (for $\epsilon \leq 0.5$) as a system with no noise added. For practical purposes in industry, our system is acceptable.

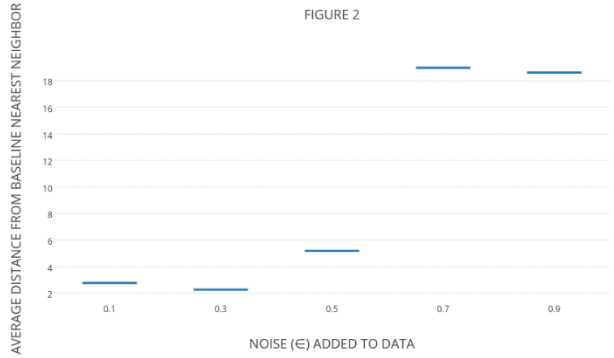

Fig. 2. **This illustrates the distance for the nearest neighbor in the baseline to the various ϵ. Each column is the average distance across all feature sizes.**

5.3. Experiment 3 Evaluation

Results from experiment 3 reinforce findings in experiments 1 and 2. Here, we look at where the highest scoring matches for various ϵ rank in the baseline similarity list.

The general results of experiment 3 show that the highest scoring matches for all $\epsilon \leq 0.5$ are one of the top 26 matches in the baseline similarity list. Figure 3 shows that for the highest scoring matches for $\epsilon \leq 0.5$, these matches are in the list of the top six matches for the baseline. This further shows that not only does the datasets with added noise ($\epsilon \leq 0.5$) keep the highest scoring matches for the baseline among their top six matches, the highest scoring matches resulting from the perturbed dataset are amongst the top six matches in the baseline.

The findings in this experiment show that the although the matches made with the noisy datasets might not be the absolute best match, they are one of the top six for $\epsilon \leq 0.5$. This means that on average, our system will match a user with one of the top six most similar users for an $\epsilon \leq 0.5$ across all feature sizes. This illustrates the practicality of our system for industry use.

FIGURE 3

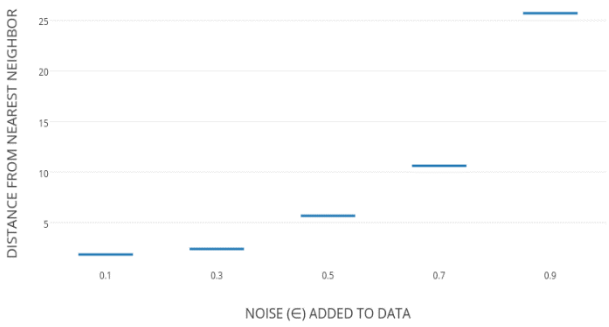

Fig. 3. **Compares where the highest scoring nearest neighbor for various ∈ ranks on the baseline similarity score list.**

5.4. Comparison of our Solution to Related Works

The results of our work shows that our system is resilient overcomes the limitations that other works have experienced. Different from [10] our work does not rely on third-party matching or computationally expensive encryption algorithms. Also, we ensure ϵ differential privacy for users data, which [8] and [14] suffer from as well. Our results show relatively accurate matching, compared to the ambiguous results of [8]. Unlike approaches that use k-anonymity, our system considers all attributes in the dataset and produces accurate matching even with some noise added to the data.

6. Conclusions and Future Work

In this work, we draw the conclusion that our matching system with differential privacy guarantees is practical, feasible, and produces highly accurate matching. Although the loss of accuracy does increase as more noise is added to the data, our system is resilient for significant levels of noise.

For matching systems used in industry, it is typically not required that the best match be made, rather, a match between two users with high similarity. We have demonstrated that our system performs well in this setting. We generally match users to one of their top six most similar neighbors while adding noise to the data. Our system does not rely third party matching schemes and allows users to set the privacy level they are comfortable with.

Directions for future work include expanding our system to work with other data sets and testing different matching protocols. Also, fully realizing our system by implementing it in an application that can be tested on mobile devices in a real world setting.

7. References

[1] C. C. Aggarwal. On k-anonymity and the curse of dimensionality. In K. Bo¨hm, C. S. Jensen, L. M. Haas, M. L. Kersten, P.-˚A. Larson, and B. C. Ooi, editors, VLDB, pages 901–909. ACM, 2005.

[2] Dwork. Differential privacy: A survey of results. In M. Agrawal, D.-Z. Du, Z. Duan, and A. Li, editors, Theory and Applications of Models of Computation, 5th International Conference, TAMC 2008, Xi'an, China, April 25-29, 2008. Proceedings, volume 4978 of Lecture Notes in Computer Science, pages 1–19. Springer, 2008.

[3] Dwork, Cynthia, and Aaron Roth. "The algorithmic foundations of differential privacy." *Foundations and Trends® in Theoretical Computer Science* 9.3–4 (2014): 211-407.

[4] E.A¨ımeur,G.Brassard,J.M.Fernandez,and F. S. M. Onana. Alambic: a privacy-preserving recommender system for electronic commerce. Int. J. Information Security, 7(5):307–334, 2008.

[5] E. A¨ımeur, G. Brassard, J. M. Fernandez, F. S. M. Onana, and Z. Rakowski. Experimental demonstration of a hybrid privacy-preserving recommender system. In ARES, pages 161–170. IEEE Computer Society, 2008.

[6] Inan, Ali, et al. "Private record matching using differential privacy." *Proceedings of the 13th International Conference on Extending Database Technology*. ACM, 2010.

[7] Korolova, Aleksandra, et al. "Releasing search queries and clicks privately." *Proceedings of the 18th international conference on World wide web*. ACM, 2009.

[8] M. Li, N. Cao, S. Yu, and W. Lou, "Findu: Privacy-preserving personal profile matching in mobile social networks," in *Proc. of IEEE INFOCOM, 2011*.

[9] McSherry, Frank, and Ilya Mironov. "Differentially private recommender systems: building privacy into the net." *Proceedings of the 15th ACM SIGKDD international*

conference on Knowledge discovery and data mining. ACM, 2009.

[10] B. Niu, X. Zhu, T. Zhang, H. Chi, and H. Li, "P-match: Priority-aware friend discovery for proximity-based mobile social networks," in *Proc. IEEE MASS*, 2013, pp. 351–355.

[11] Shen, Yilin, and Hongxia Jin. "Privacy-preserving personalized recommendation: An instance-based approach via differential privacy." *Data Mining (ICDM), 2014 IEEE International Conference on*. IEEE, 2014.

[12] R. Zhang, Y. Zhang, J. Sun, and G. Yan, "Fine-grained private matching for proximity-based mobile social networking," in *Proc. of IEEE INFOCOM, 2012*.

[13] Zhang, Rui, et al. "Privacy-preserving profile matching for proximity-based mobile social networking." *IEEE Journal on Selected Areas in Communications* 31.9 (2013): 656-668.

[14] X. Zhu, J. Liu, S. Jiang, Z. Chen, and H. Li, "Efficient weight-based private matching for proximity-based mobile social networks," in *Proc. IEEE ICC*, 2014, pp. 4114–4119.

[15] Zhu, Xiaoyan, Zengbao Chen, and Wenye Wang. "G2G: Privacy-preserving group matching for proximity-based mobile social networks." *Communications in China (ICCC), 2015 IEEE/CIC International Conference on*. IEEE, 2015.

[16] https://www.kaggle.com/miroslavsabo/young-people-survey