Privacy-Preserving User Profiling with Facebook Likes

Sanchya Bhagat[1], Keerthanaa Saminathan[1], Anisha Agarwal[1],
Rafael Dowsley[2], Martine De Cock[1,3], Anderson Nascimento[1]
[1]School of Engineering and Technology, University of Washington, Tacoma, USA
Email: {sanchya, keergs, anisha3, mdecock, andclay}@uw.edu
[2] Dept. of Computer Science, Aarhus University, Aarhus, Denmark, Email: rafael@cs.au.dk
[3] Dept. of Applied Math., Comp. Sc. and Statistics, Ghent University, Email: martine.decock@ugent.be

I. Introduction

The content generated by users on social media is rich in personal information that can be mined to construct accurate user profiles, and subsequently used for tailored advertising or other personalized services. Facebook has recently come under scrutiny after a third party gained access to the data of millions of users and mined it to construct psychographical profiles, which were allegedly used to influence voters in elections. As part of a possible solution to avoid data breaches while still being able to perform meaningful machine learning (ML) on social media data, we propose a privacy-preserving algorithm for k-nearest neighbor (kNN) [1], one of the oldest ML methods, used traditionally in collaborative filtering recommender systems.

In our approach, which is based on Secure Multiparty Computation (SMC) [2], 1000s of users each send encrypted shares of their data to two non-colluding service clouds, nicknamed *Alice* and *Bob* in Fig. 1. When a new instance for user *Carol* has to be classified, it is split into encrypted shares and sent to *Alice* and *Bob*, who subsequently engage in a secure kNN protocol. In the end, *Alice* and *Bob* each hold a share of the final result. They disclose the shares to the user and/or the advertisement server on the social media platform, which can use it to decide which advertisements to display. Throughout this process, none of the users sees the data of any of the other users in an unencrypted way. In addition, the computational servers *Alice* and *Bob* never see the unencrypted data from *Carol* nor from any of the 1000s of training users.

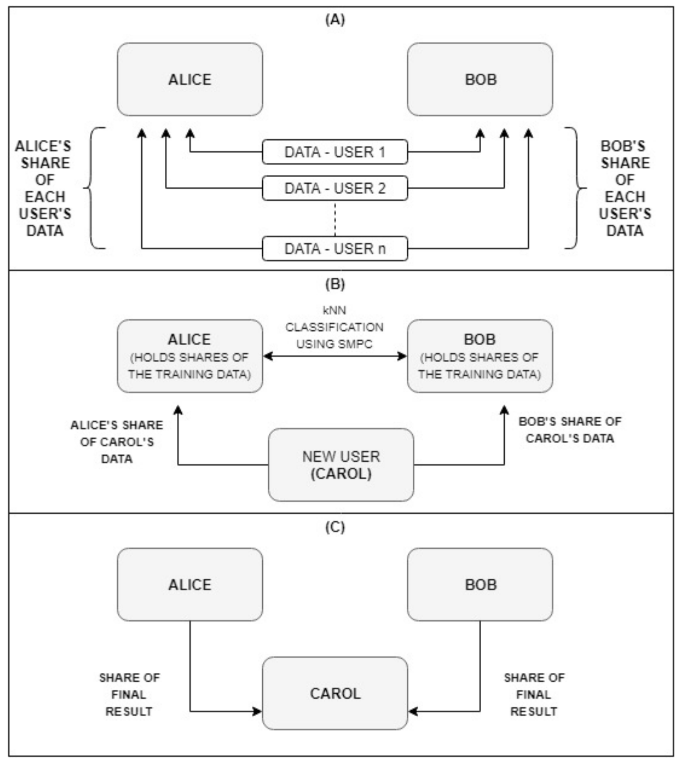

Fig. 1. Architecture of the two-server model for secure kNN

II. Related Work

Given the popularity of kNN in machine learning and data mining, it is not surprising that efforts have already been made to perform kNN in a privacy-preserving manner. Our goal is to keep both the training data and the query instance private, unlike existing methods that assume that the query point is publicly known [3], [4], or that leak which instances are among the k nearest neighbors [5]. Unlike differential privacy based methods that trade accuracy for privacy [6], [7], our protocols produce the exact same outcome and accuracy as in the clear. Furthermore, our interest is in scenarios where the original training data is owned by 1000s of users instead of the scenario where all training data is owned by one party, as in [8], [9] and most of [10].

Rane and Boufounos [10] do provide a sketch on how to perform privacy-preserving 1NN in the multi-party case (as we consider) based on the use of *Shamir's secret sharing* as described in [2]. We use *additive secret sharing* instead, because it is computationally cheaper and has better round complexity, i.e. a lower number of sequential steps in the protocol (discounting on operations that can be done in parallel). The main differences and improvements of our work over that of Rane and Boufounos [10] are that we propose a protocol for privacy-preserving kNN – which is considerably more challenging than 1NN – and that we go well beyond a mere sketch by presenting an implementation and experimental

TABLE I
TIME TO PRIVATELY CLASSIFY A NEW INSTANCE, AND CLASSIFICATION ACCURACY MEASURED OVER 1500 TEST INSTANCES

k	Training Data Set Size	Runtime (min) Sort-and-Swap	Runtime (min) Threshold-kSelect	Accuracy
16	500	19.81	2.79	66.41
16	5000	180.88	31.98	68.40
16	8000	345.03	50.69	66.04
32	500	36.24	2.80	65.82
32	5000	285.44	32.29	67.79
32	8000	454.01	50.63	69.84

evaluation of the protocols, which allows us to measure runtimes.

III. METHODS AND RESULTS

We use data of 9500 Facebook users and 600 items (pages) collected in the myPersonality project [11]. For each of the users $\mathbf{u}$ and each of the items i, the dataset contains information on whether user $\mathbf{u}$ has clicked on the like button for item i, i.e. $\mathbf{u}(i) = 1$ or not, i.e. $\mathbf{u}(i) = 0$. "Likes" information can be used to infer all kinds of user characteristics, including personality, and demographics (see e.g. [12], [13]). In our experiments we use it to derive the gender of the user. The reported runtime results are similar for any other label that one might be interested in to derive.

To measure the proximity of users, we use the Jaccard distance, which is an appropriate metric to use in kNN when dealing with sparse data such as likes information. We designed a cryptographic protocol that allows *Alice* and *Bob* from Fig. 1 to securely compute the numerator and the denominator of the Jaccard distance between 2 users for which they have shares (such as the data of new user *Carol* and the data of any training user). Using this protocol for secure computation of the Jaccard distance, we developed and implemented two different algorithms that allow *Alice* and *Bob* to infer the class label for a new user with kNN in a privacy-preserving manner.

The first algorithm combines the Jaccard distance computation with an algorithm for oblivious sorting [14] to sort the first k elements. We then use an oblivious swap circuit to iterate through the rest of the training examples and swap them with the first k elements wherever required thereby ensuring that the first k distances are the smallest distances among all the training examples. We call this method *Sort-and-Swap kNN*.

The second algorithm is based on the algorithm for finding the top k elements proposed by Vaidya and Clifton [15] (using Yao's secure comparison) and used by Burkhart and Dimitropoulos [16] (based on Shamir' secret sharing scheme). This algorithm finds the k nearest elements without requiring sorting at all, namely by iteratively looking for a cut-off point through doing binary search among the distances. We call this method *Threshold-kSelect kNN*.

We implemented the Sort-and-Swap and Threshold-kSelect algorithms above in the SMC framework Lynx.[1] The reported runtimes reported in Table I are the average of 3 executions, run in two AWS EC2 instances (*Alice* and *Bob*) with 36 vCPUs and 72 GB RAM.

The Threshold-kSelect kNN algorithm, which does not involve any sorting at all, is significantly faster than the Sort-and-Swap algorithm. Moreover, unlike for the Sort-and-Swap algorithm, increasing the value of k does not cause a significant increase in the runtimes of the Threshold-kSelect kNN algorithm. Even though the asymptotic complexity of Sort-and-Swap kNN is $\mathcal{O}(kn)$ and the complexity of threshold-kSelect kNN is $\mathcal{O}(n \log n)$, the latter is faster as it involves independent microservices running in parallel. This is not the case with Sort-and-Swap kNN as each swap operation has to happen sequentially.

[1] https://bitbucket.org/uwtppml/lynx

REFERENCES

[1] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[2] R. Cramer, I. Damgaard, and J. B. Nielsen, *Secure multiparty computation and secret sharing*. Cambridge University Press, 2015.

[3] M. Kantarcıoǧlu and C. Clifton, "Privately computing a distributed k-nn classifier," in *Proc. of the 8th European Conference on Principles of Data Mining and Knowledge Discovery*, ser. Lecture Notes in Artificial Intelligence, vol. 3202. Springer, 2004, pp. 279–290.

[4] H. Polat and W. Du, "Privacy-preserving top-n recommendation on horizontally partitioned data," in *Proc. of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, 2005, pp. 725–731.

[5] M. Shaneck, Y. Kim, and V. Kumar, "Privacy preserving nearest neighbor search," in *Machine Learning in Cyber Trust*. Springer, 2009, pp. 247–276.

[6] F. McSherry and I. Mironov, "Differentially private recommender systems: Building privacy into the Netflix prize contenders," in *Proc. of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 627–636.

[7] M. Hou, R. Wei, T. Wang, Y. Cheng, and B. Qian, "Reliable medical recommendation based on privacy-preserving collaborative filtering," *Computers, Materials & Continua*, vol. 56, no. 1, pp. 137–149, 2018.

[8] B. K. Samanthula, Y. Elmehdwi, and W. Jiang, "K-nearest neighbor classification over semantically secure encrypted relational data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 5, pp. 1261–1273, 2015.

[9] H.-J. Kim, H.-I. Kim, and J.-W. Chang, "A privacy-preserving knn classification algorithm using Yao's garbled circuit on cloud computing," in *10th International Conference on Cloud Computing (CLOUD)*, 2017, pp. 766–769.

[10] S. Rane and P. T. Boufounos, "Privacy-preserving nearest neighbor methods: Comparing signals without revealing them," *IEEE Signal Processing Magazine*, vol. 30, no. 2, pp. 18–28, 2013.

[11] M. Kosinski, S. C. Matz, S. D. Gosling, V. Popov, and D. Stillwell, "Facebook as a research tool for the social sciences: Opportunities, challenges, ethical considerations, and practical guidelines." *American Psychologist*, vol. 70, no. 6, p. 543, 2015.

[12] W. Youyou, M. Kosinski, and D. Stillwell, "Computer-based personality judgments are more accurate than those made by humans," *Proceedings of the National Academy of Sciences*, vol. 112, no. 4, pp. 1036–1040, 2015.

[13] G. Farnadi, G. Sitaraman, S. Sushmita, F. Celli, M. Kosinski, D. Stillwell, S. Davalos, M.-F. Moens, and M. De Cock, "Computational personality recognition in social media," *User modeling and user-adapted interaction*, vol. 26, no. 2-3, pp. 109–142, 2016.

[14] D. Bogdanov, S. Laur, and R. Talviste, "Oblivious sorting of secret-shared data," Technical Report T-4-19, Cybernetica, http://research.cyber.ee, Tech. Rep., 2013.

[15] J. Vaidya and C. Clifton, "Privacy-preserving top-k queries," in *Proc. of 21st International Conference on Data Engineering*, 2005, pp. 545–546.

[16] M. Burkhart and X. Dimitropoulos, "Fast privacy-preserving top-k queries using secret sharing," in *19th International Conference on Computer Communications and Networks*, 2010.