SaDes: An Interactive System for Sensitivity-aware Desensitization towards Tabular Data

Kechun Zhao
zkc0422@outlook.com
School of Cyber Engineering
Xidian University
Xi'an, China

Hui Li
hli@xidian.edu.cn
State Key Laboratory of Integrated Services Networks
Xidian University
Xi'an, China

Zheng Gong
marcogong22@gmail.com
School of Cyber Engineering
Xidian University
Xi'an, China

Jiangtao Cui
School of Computer Science and Technology
Xidian University
Xi'an, China
cuijt@xidian.edu.cn

ABSTRACT

Before the publication of particular datasets, in order to protect the private information while preserving the usability as much as possible, desensitization is required. Automatic identification and evaluation of sensitive attributes are prerequisites for targeted desensitization of datasets, sensitivity can also reflect the effect of desensitization in turn. However, existing desensitization systems all rely on predefined desensitization model with respect to manually given sensitivity levels, which is subjective and unable to be applied end-to-end. Besides, there is no way for the user to tell whether the desensitization is performed enough or superfluous. In this demonstration, we present an interactive system for sensitivity-aware desensitization towards tabular data (SaDes). It automatically evaluates the risks of re-identification for arbitrary columns according to record-linkage attack, and performs desensitization accordingly. The risks of re-identification for the desensitized data can be immediately evaluated such that the user can iteratively execute desensitization in order to achieve a better balance between the usability and privacy. To the best of our knowledge, SaDes is the first system that provides automatic sensitivity evaluation and interactive desensitization in a back-to-back manner.

CCS CONCEPTS

• **Security and privacy → Data anonymization and sanitization**; *Pseudonymity, anonymity and untraceability*; • **Theory of computation → Theory of database privacy and security**.

KEYWORDS

data security, data desensitization, sensitivity quantification, record-linkage attack

CIKM '21, November 1–5, 2021, Virtual Event, Australia.

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3481975

ACM Reference Format:
Kechun Zhao, Hui Li, Zheng Gong, and Jiangtao Cui. 2021. SaDes: An Interactive System for Sensitivity-aware Desensitization towards Tabular Data. In *Proceedings of the 30th ACM Int'l Conf. on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3459637.3481975

1 INTRODUCTION

Nowadays, the collection, mining [9] and analysis of datasets [1] [4] from multiple stakeholders have shown great commercial value. However, before publication or sharing, data owners have to perform some desensitization tasks such that the sensitive information in the dataset can be protected against potential adversaries. Generally, the desensitization refers to the transformation of sensitive information through a series of predefined rules, to balance data privacy and usability. Nevertheless, as the correlation between datasets increases, it is easier for adversaries to collect other datasets as background knowledge, from which the sensitive information can be inferred from record-linkage attack [6, 10]. In this situation, before publishing/sharing the dataset, it is vital for the data owners to judge whether a particular column is sensitive than another, or to what extent a column can be easily used to infer the record through potential background knowledge [7].

Existing data desensitization systems such as Oracle Data Masking[1], DMS of DBSEC[2], and Alibaba data security center[3], focus on addressing specific desensitization requirements in various applications, but with a series of limitations. For instance, Alibaba data security center requires users to manually mark which sensitive attribute needs to be desensitized. Oracle Data Masking, DMS and Privitar[4] predefine regular expressions to detect some semantically sensitive attributes and generalization rules for desensitization. However, it is impossible to predefine templates for all attributes we may meet with and the format of the values in the same column may be diverse, even if they exhibit the same semantic meaning. Moreover, recognizing sensitive attributes only from the semantic level ignores the interaction between data. Besides, because different attributes have diverse characteristics, it is not appropriate

[1] https://www.oracle.com/database/technologies/security/data-masking-subsetting.html
[2] https://www.dbsec.cn/pro/dms-s.html
[3] https://help.aliyun.com/product/88674.html
[4] https://www.privitar.com/publisher

to blindly apply a unified desensitization operation to the whole dataset.

In order to 'safely' publish/share the dataset without worrying about the leakage risk of sensitive information, in this demonstration, we present an interactive system for sensitivity-aware desensitization towards tabular data (SaDes), which combines automatic quantification of sensitive attributes and sensitivity-guided desensitization to provide a back-to-back iterative sensitivity-driven desensitization service.

2 SaDes SYSTEM

The system can be divided into two modules: sensitivity quantification and desensitization. The former automatically quantifies the sensitivity of columns and provides guidelines for the desensitization operations adopted by the latter. After the desensitization operations are finished, the former can also be used as metrics to tell whether the desensitization operations satisfy the user's privacy protection requirements or not.

2.1 Sensitivity Quantification

The dataset sensitivity quantification refers to the quantitatively evaluate the sensitivity of the attributes in the targeted dataset. As the first step of SaDes, we adopt and implement our latest solution in [2], which calculates the probability of each attribute in the dataset being attacked successfully, and use this probability as the sensitivity of attribute. Our model can quantify the sensitivity of the columns no matter the semantics of columns are known or not.

Adversary Model. According to Record-linkage Attack[8], for the multivariate tabular dataset, an adversary could link personally identifiable attributes (eg, social security number), or quasi-identifier attributes (eg, age, gender), which may not expose the identity of the record independently but can leak the information if combined with other attributes, with his background knowledge to identify a particular or group of individuals in the dataset. To formally define the adversary model, we adopt the concept of minimal unique column combination in [3], denoted as *UCC*. Specifically, the set of *UCC* is the set of independent primary keys and composite primary keys in a relational database *R*. Assume that the adversary has background knowledge κ, with which Record-linkage Attack can be performed to infer the real identification of some persons *i* in *R*. Obviously, as long as attribute(s) in κ can constitute a *UCC* of *R*, the adversary would successfully identify *i*, which is recognized as a successful attack. On the contrary, if κ doesn't contain any *UCC*, the adversary can never know if any information he got certainly corresponds to a unique tuple *i* in *R*.

The Risk of Re-identification. Given a database instance *R* with *n* attribute columns, denoted as $A_1, \ldots, A_n$, for each column combination *U* over them (it can be viewed as an extensive projection without eliminating duplicate rows), if some rows of *U* are contained in κ, we denote them by $U \ltimes \kappa$. Suppose the probabilities for $\{A_i\} \ltimes \kappa$ are independent with each other and referred to as $p(A_i)$, respectively, then the success rate for Record-linkage Attack with respect to a particular column can be carried out as follows.

Firstly, if a column A_i itself constructs a *UCC*, the probability of successful attack through A_i should be $p(A_i)$. The reason is that as long as some rows of these columns are contained in κ, the adversary can definitely execute the attack. Secondly, if A_i is an element of some *UCC*s but not a *UCC* itself, the probability of successful attack through it would also depend on other columns that appear in those *UCC*s. Generally, we denote these *UCC*s as $U(A_i) = \{UCC | A_i \in UCC\}$. For each $UCC_j \in U(A_i)$, the adversary needs to cover other columns to successfully implement the attack once A_i is revealed. Suppose the columns that appear along with A_i in UCC_j is $B_1, ..., B_k$, let $P(UCC_j \ltimes \kappa)$ be the probability for UCC_j to be revealed, then its posterior probability, given that A_i is exposed, can be computed as $P(UCC_j \ltimes \kappa | \{A_i\} \ltimes \kappa) = \prod_{r=1}^{k} p(B_r)$. Notably, we also have to take into account the success rate for attacking through other *UCC*s in $U(A_i)$. Given $U(A_i)$, the adversary will successfully complete the attack if $\exists UCC_j \in U(A_i)$ such that $UCC_j \ltimes \kappa$. Therefore, once A_i is exposed, the successful attack probability through any *UCC* that contains A_i would be

$$P(success | \{A_i\} \ltimes \kappa) = 1 - \prod_{UCC_j \in U(A_i)} (1 - P(UCC_j \ltimes \kappa | \{A_i\} \ltimes \kappa)) \tag{1}$$

Equation 1 in fact evaluates the posterior probability given that A_i has been exposed. Then the eventual probability for successful attack via A_i should be

$$S(A_i) = p(A_i) P(success | \{A_i\} \ltimes \kappa). \tag{2}$$

Notably, according to Equation 2, $S(A_i)$ depends on $p(A_i)$, which refers to the general probability for A_i being revealed to an arbitrary adversary. Obtaining those probabilities seems to be a challenging task. For ease of discussion, we set $p(A_i)$ uniformly as 0.5 in the followings such that we can focus on discussing the intrinsic distribution-driven characteristics of each column, excluding all external scenario-dependent factors.

2.2 Sensitivity-guided Desensitization

Given the sensitivity automatically computed from the previous step, we mainly use two data desensitization methods, generalization and masking, in SaDes, depending on the type of the attribute. Data generalization [5] is mainly used for the desensitization of common sensitive attributes that can be identified through pre-defined rules, *e.g.,* ID number, date, sex, etc. The original entry for each tuple in the column is modified with coarser granularity by generalization. The generalized value reflects the characteristics of the original value of the same entry but hiding the details. Data masking is mainly used to desensitize numerical attributes and the attributes with unknown semantics. We use '★' as a mask to replace a specific length of digits in the original value.

Therefore, given a tabular dataset to be desensitized, we have to first test each column against a series of pre-defined rules in order to tell the semantics of the columns. After that, the columns can be classified into two groups, one with semantics but not numerical, and otherwise. For the first group, depending on the semantic meaning of the column, the corresponding generalization schemes vary. For both groups, the degree of desensitization/masking also vary and is guided by the sensitivity acquired from the first module. Altogether there are 6 generalization levels, which we determine by the conceptual induction levels of common sensitive data, as shown in Figure 1. For instance, for the column identified as *Name*, we divide the name into last and first ones. The desensitization method we define for *Address* attribute is constructed according to concept hierarchies, *e.g.,* state-city-district-street. For the second group of columns, we apply data masking as follows, we divide the

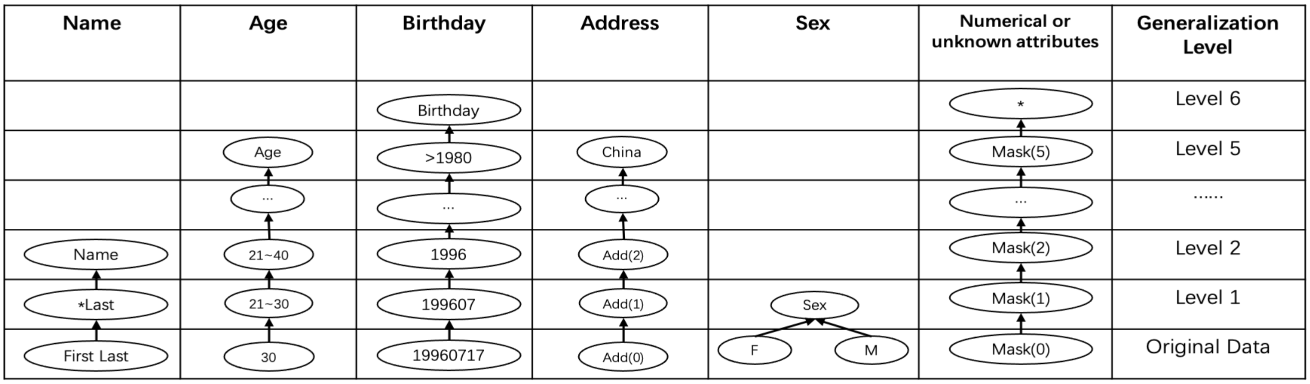

Figure 1: Desensitization Methods

value length into six segments with equal width and incrementally mask one segment as the desensitization level goes up.

For both methods, a lower desensitization level refers to a lower degree of privacy protection, *i.e.,* a better usability of the desensitized dataset. According to SaDes, the dataset after the highest level of desensitization may share the same value for each attribute. Therefore, the risk of re-identification will drop to 0, that is, the sensitivity (computed according to Equation 2) of desensitized dataset is 0, which in turn justifies the rationality of our sensitivity evaluation model. Combined with the requirements for the desensitized dataset, the user can decide whether to choose a higher or lower level of desensitization operation according to the sensitivity of the desensitized dataset.

3 SYSTEM DESIGN

3.1 Effectiveness of the System

In the demonstration, we shall provide a series of datasets for the audience to choose, hereby we shall use RPI to illustrate the effect of SaDes. RPI is a personal information tabular table containing 14 columns and 6478 tuples, collected from a bank who is collaborating with us on sensitivity study. These columns semantically refer to user ID number, birthday, address, mobile number (abbr., Hp.), gender, etc. Notably, the values in Name column are faked ones.

Table 1: the UCCs of RPI

UCC	Column combinations	UCC	Column combinations
UCC_1	*Id*	UCC_9	*Address,Hp.,Name,Zip*
UCC_2	*CtfId*	UCC_{10}	*Birthday,District3,Hp.*
UCC_3	*Birthday,Hp.,Zip*	UCC_{11}	*Birthday,District4,Hp.*
UCC_4	*Gender,Hp.,Name*	UCC_{12}	*Birthday,Hp.,Name*
UCC_5	*Address,Hp.,Name,Tel*	UCC_{13}	*Birthday,CtfTp,Gender,Hp.*
UCC_6	*Address,District3,Hp.,Name*	UCC_{14}	*Birthday,Gender,Hp.,Tel*
UCC_7	*Address,District4,Hp.,Name*	UCC_{15}	*Address,Birthday,Hp.*
UCC_8	*Address,Fax,Hp.,Name*		

For ease of understanding, the *UCC*s of RPI are listed in Table 1, we *uniformly* set the reveal probabilities for columns as 0.5. Figure 2 shows sensitivity results for each column according to Equation 2. *ID* and $CtfID$ exhibit the highest sensitivity. In fact, either *ID* or $CtfID$ can construct a *UCC* itself as shown in Table 1. For $Birthday$, an attack is successful only when the adversary gets other columns in *UCC*s that contain $Birthday$, such as $Mobile$ and Zip of UCC_3 in Table 1. Since the attack on $Birthday$ requires more information, it would be more difficult to succeed when compared with the attacks on *ID* and $CtfID$. Naturally, its sensitivity should be lower than them, which is justified in Figure 2.

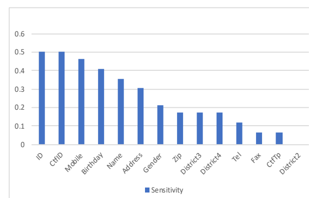

Figure 2: Sensitivity of RPI

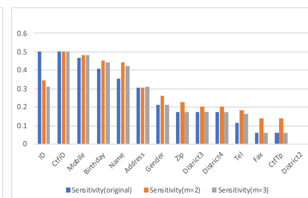

Figure 3: Sensitivity of RPI after desensitization

To further illustrate the effectiveness, we use data masking method to desensitize columns with high sensitivity, and re-compute the sensitivity afterwards. For *ID*, we separately mask the the last m digits of each entry with '$\star$'. Figure 3 shows the sensitivity computed before and after the masking. Obviously, a higher level desensitization for *ID* results in a lower sensitivity, which also justifies the rationality of our sensitivity evaluation scheme. Notably, the sensitivity of *ID* has dropped significantly, respectively; the sensitivity of some other columns increase accordingly. Due to the masking, *ID* is no longer a key in the table, and forms new UCCs together with other attributes. Consequently, for the attributes of newly-formed UCCs containing the newly masked *ID*, their sensitivities will increase; for the rest columns, their sensitivities will remain unchanged. Experiments show that the computed sensitivity results not only conform to our subjective judgments, but also reflect the effect of desensitization in real-time.

3.2 GUI Components

The GUI of SaDes consists of two panels (tabs). The first panel is the desensitization configuration interface, which consists of 4 components (C1-C4 in Figure 4). SaDes extracts each attributes of the original dataset, evaluates the corresponding sensitivity and displays the results accordingly in C1. According to the sensitivity in C1 and desensitization requirements, the user can choose whether to desensitize each attribute and which desensitization method to use in C2. C3 enables the user to tune the intensity of the desensitization on the dataset which varies from Level-1 to Level-6. After clicking the start button in C3, SaDes will apply corresponding desensitization operations accordingly. Then, the sensitivity of

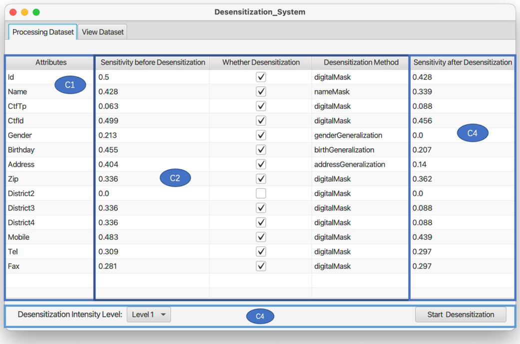

Figure 4: SaDes: panel 1

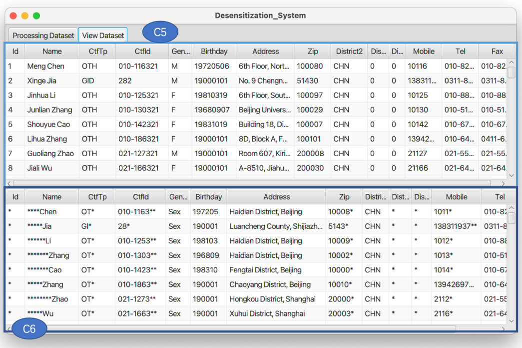

Figure 5: SaDes: panel 2

each attribute after desensitization will be immediately recomputed and displayed in C4. The second panel (Figure 5) compares and displays the values in the dataset, where C5 shows the original values, and C6 shows the desensitized ones accordingly.

4 DEMONSTRATION PLAN

Due to privacy issue, we select to implement SaDes as a desktop application but not a web one, which inevitably introduces risk of leakage over the web server. Currently, SaDes supports input formats in CSV, etc. The purpose of our demonstration is to enable the audience experience the diverse interactive features of SaDes. Our demonstration will provide a series of real-world dataset including the aforementioned RPI, where the entries for the *Name* column in all the datasets are faked ones. A **short video** to illustrate the main features of SaDes using example use cases is available at https://youtu.be/iitSb8NxGzU.

Through the GUI, the audience can browse all the attributes of the dataset (*e.g.,* RPI) and their corresponding sensitivity in C1, which is shown in the first two columns of Figure 6. According to the sensitivity and requirements for the usage of the desensitized dataset, the audience can select the corresponding desensitization operation. For common sensitive attributes, SaDes adopts corresponding predefined generalization methods by default in C2. For attributes that have unknown semantics or fail to fit in any predefined desensitization methods, *e.g.,* District3 and District4, SaDes can automatically switch from generalization to data masking. For

Attributes	Sensitivity before Desensitization	Sensitivity after Desensitization	Sensitivity after Desensitization	Sensitivity after Desensitization
Id	0.5	0.428	0.204	0.0
Name	0.428	0.339	0.0	0.0
CtfTp	0.063	0.088	0.0	0.0
CtfId	0.499	0.456	0.285	0.0
Gender	0.213	0.0	0.0	0.0
Birthday	0.455	0.207	0.114	0.0
Address	0.404	0.14	0.031	0.0
Zip	0.336	0.362	0.031	0.0
District2	0.0	0.0	0.0	0.0
District3	0.336	0.088	0.0	0.0
District4	0.336	0.088	0.164	0.0
Mobile	0.483	0.439	0.283	0.0
Tel	0.309	0.297	0.23	0.0
Fax	0.281	0.297	0.192	0.0

Figure 6: The Sensitivity of RPI Before and After Different Level of Desensitization

Id	Name	CtfTp	CtfId	Ge...	Birthday	Address	Zip	Distr...	Dis...	Di...	Mobile	Tel	Fax
*	Name	*	021-******	Sex	1980~1990	Shanghai	201***	CHN	*	*	21***	021-691******...	021-69******
*	Name	*	021-******	Sex	1970~1980	Shanghai	200***	CHN	*	*	21***	021-64******	021-644****...
*	Name	*	027-******	Sex	1980~1990	Hubei Provi...	430***	CHN	*	*	27***	027-87******	027-87******
*	Name	*	029-******	Sex	1970~1980	Shanxi Provi...	710***	CHN	*	*	29***	029-876*****...	029-876****...
*	Name	*	0510-******	Sex	1960~1970	Jiangsu Pro...	214***	CHN	*	*	510***	0510*********	0510*********
*	Name	*	0531-******	Sex	1980~1990	Shandong P...	250***	CHN	*	*	531***	0531-8296**...	0531*********
*	Name	*	0571-******	Sex	1980~1990	Zhejiang Pr...	311***	CHN	*	*	571***	0571*********	0571*********
*	Name	*	0571-******	Sex	<1940	Zhejiang Pr...	310***	CHN	*	*	571***	0571*********	0571*********
*	Name	*	0571-******	Sex	<1940	Zhejiang Pr...	310***	CHN	*	*	571***	0571*********	0571*********

Figure 7: RPI After Level-3 Desensitization

the attribute District2 with sensitivity of 0, the audience can select not to apply any desensitization on it.

The audience are recommended to experience at least three levels of desensitization in the demo. Firstly, they can choose to apply the lowest desensitization level in C3, after which the audience can observe the sensitivity after desensitization in C4 (the third column of Figure 6), while the values before and after desensitization are shown in C5 and C6. If the desensitized dataset does not satisfy the expected privacy requirements, the audience can change the desensitization configuration in C2 and C3, and desensitize the dataset again by clicking the start button. The audience can then apply Level-2 desensitization. At this time, the sensitivity of each attribute has been greatly reduced compared to Level-1, which is shown in the forth column of Figure 6. The sensitivity of some attributes has been reduced to 0, which indicates that there is negligible risk of privacy leakage.

To completely eliminate the risk of privacy leakage, the audience can further apply higher level of desensitization. As shown in Figure 7, although the sensitivity is 0 (the last column in Figure 6), the usability of the dataset after desensitization has not been completely eliminated. For example, the attribute *Address* still retains the location information, and the attribute *Birthday* retains the age range information. SaDes enables the user to iteratively perform sensitivity quantification and sensitivity-guided desensitization in a back-to-back manner, where the sensitivity changes caused by the desensitization are shown in real-time, until a satisfied trade-off between usability and privacy is achieved.

ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China (No. 61972309), CCF-Huawei Database System Innovation Research Plan (No. 2020010B), Key Scientific Research Program of Shaanxi Provincial Department of Education (No. 20JY014).

REFERENCES

[1] Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2020. TURL: Table Understanding through Representation Learning. *PVLDB* 14, 3 (2020), 307–319.

[2] Zheng Gong, Kechun Zhao, Hui Li, and Yingxue Wang. 2020. Instance-Aware Evaluation of Sensitive Columns in Tabular Dataset. In *Web and Big Data - 4th International Joint Conference, APWeb-WAIM 2020, Tianjin, China, September 18-20, 2020, Proceedings, Part I*. 11–19.

[3] Arvid Heise, Jorge-Arnulfo Quiané-Ruiz, and Ziawasch Abedjan. 2013. Scalable Discovery of Unique Column Combinations. *PVLDB* 7, 4 (2013), 301–312.

[4] Tuan-Anh Hoang, Khoi Duy Vo, and Wolfgang Nejdl. 2018. W2E: A Worldwide-Event Benchmark Dataset for Topic Detection and Tracking. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*. ACM, 1847–1850.

[5] Florian Kohlmayer, Fabian Prasser, Claudia Eckert, Alfons Kemper, and Klaus A. Kuhn. 2012. Flash: Efficient, Stable and Optimal K-Anonymity. In *2012 International Conference on Privacy, Security, Risk and Trust(PASSAT) and 2012 International Confernece on Social Computing*. IEEE, 708–717.

[6] Liu Kun, Das Kamalika, Gr Tyrone, and Kargupta Hillol. 2008. *Privacy-Preserving Data Analysis on Graphs and Social Networks.* CRC Press / Chapman and Hall, New York.

[7] Arvind Narayanan and Vitaly Shmatikov. [n.d.]. De-anonymizing Social Networks. In *30th IEEE Symposium on Security and Privacy (S & P)*.

[8] Latanya Sweeney. 2002. k-Anonymity: A Model for Protecting Privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10, 5 (2002), 557–570.

[9] Ruijie Wang, Yuchen Yan, Jialu Wang, Yuting Jia, Ye Zhang, Weinan Zhang, and Xinbing Wang. 2018. AceKG: A Large-scale Knowledge Graph for Academic Data Mining. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*. ACM, 1487–1490.

[10] Gilbert Wondracek, Thorsten Holz, Engin Kirda, and Christopher Kruegel. 2010. A Practical Attack to De-anonymize Social Network Users. In *31st IEEE Symposium on Security and Privacy (S & P)*. IEEE, 223–238.