Investigation of Temporal De-anonymization and Practical Solution for Anonymization

https://github.com/akmami/Temporal-De-anonymization

Lara Merdol, Berke Ceran, Akmuhammet Ashyralyyev, Atakan Dönmez

December 27, 2022

Abstract—**Social media sites such as Facebook, LinkedIn, and Snapchat have been observed to collect enormous data from their users. They share some of the data with the public for research purposes. Unfortunately, this raises privacy concerns due to the probability of profile matching. However, it is also important to investigate temporal data and its effect on the risk of linking anonymous profiles to real identities.**

In this paper, we investigated the methods that are being used for anonymization and their effectiveness on temporal data. We showed the insufficiency of k-anonymity, data masking, and data swapping and proposed our method for the anonymization of data. Also, we were able to develop an efficient and accurate privacy risk quantification framework for temporal data.

In order to demonstrate the performance of our method, we generated the time-varying data and applied our method to it. The results show that our solution performs better than the other ones.

Index Terms—**temporal data, profile matching, anonymization, and de-anonymization.**

I. Introduction

As finding and storing data becomes more accessible, the variety and the number of personal data exponentially increases [1]. With the help of the advancement of information technology, a growing amount of personal data is being gathered, used, shared, and distributed [2]. Despite providing utility and functionality, users' privacy has been severely affected.

The widespread use of our personal information to personalize experiences, increase sales, and maximize returns also has an impact on the flow of ideas and the global financial system [2]. These disruptive forces have a real impact on people's constitutional rights, such as the judicial process, the right to appeal, freedom of speech, the right to vote, and more. The influence of the data analytic firm Cambridge Analytica over the US elections is one instance that demonstrates the strength of these factors [3]. By acquiring illicit access to the personally identifiable information of more than 87 million voters provided by Facebook, Cambridge Analytica gained the ability to "micro-target" specific customers or voters with messages most likely to change their behavior.

Another important concern is related to the regular publishing of data by social media companies for research purposes. For example, Facebook published some portion of its network in graph representation where the nodes are anonymous users and edges are the connections between users [4]. Despite the anonymization techniques, it is shown that the graph can be de-anonymized with high accuracy [5]. The literature definition of this event is profile matching, which is the event of removing the anonymity of an anonymous person by matching their identity with their activities on the internet, their friends, or the environments they are in, revealing private information about that person and deciphering who that person is [6]. Some mitigation techniques are proposed against such attacks but the risk of de-anonymization remains high.

On the other hand, the significant changes in time-varying graphs can increase the risk of de-anonymization or profile matching. These changes are defined as temporal data and pregnancy, cancer, etc. can be given as an example. These identifiers can link back to the user's identity if such changes are drastic or rare in the graph.

There is no technique available for temporal data besides the well know anonymization techniques. Those techniques are implemented on the principle of k-anonymity, l-diversity, and t-closeness. Due to the practicability concerns, mainly k-anonymity is used for the anonymization of graphs alongside other techniques.

In this paper, we introduced the concepts and provided background information in section II. Also, we investigated the 3 main techniques under k-anonymity on the temporal data under specific constraints in the same section and proved their inadequacy. In section III, we provided a practical solution for temporal data anonymization along with proof of satisfaction with the utility of the dataset and the anonymity of the users. Our solutions were tested by the simulated data similar to the social media graphs and their performance was discussed in sections IV and V.

II. Background

In this section, we provide a brief introduction about temporal data and social media sites. Before going through the anonymization part, we define the constraints and attacking models since we cannot investigate all the scenarios. For the anonymization of the data published by social media websites, we investigated 3 main techniques which are widely used and proved their inadequacy for each of them.

A. Model and Definitions

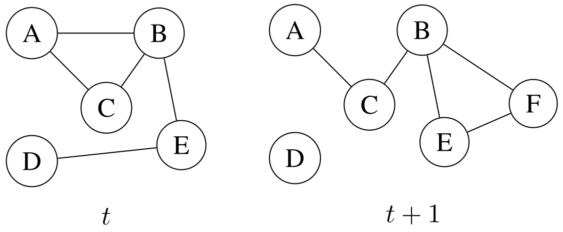
Throughout this paper, we use the following models and definitions to describe our anonymization technique to mitigate de-anonymization by temporal data.

Social Network: A social network S is modeled as a graph $G = (V, E)$ where nodes and edges correspond to V and E. Each node represents users in the social media sites and E represents the connection between users. The definition of connection can vary from one social media website to another, e.g. Facebook uses friends, LinkedIn uses connections, etc. but for simplicity's sake we used connection. Each node contains multiple edges to different nodes and it contains information about the users, i.e. identifiers, which are processed by the anonymization technique before being published.

For the simplicity of demonstration, we classified each attribute to its specific column. As an example, each node will have the same number of attributes, even possibly null, and for some of the attributes, it can be multiple values denoted as a list.

Time-Varying Graph: Time-varying datasets are the datasets that are being published regularly with similar content differing only in minor attributes. Especially, social media companies release the entire or some portion of their graphs, after an anonymization process, to the public for research purposes [7]. There is no specific time interval set for publishing and it is totally up to the company when and how frequently it will be done.

The changes between graphs can be either or both in nodes' attributes and edges. The difference between graphs' shapes can result from either network/connection changes between users, such as making new connections or removing an old one, or deliberately being manipulated before releasing for anonymization processes. Due to the high similarity between time-varying graphs, the de-anonymization risks increase with the correlation between them over time which results in deliberate manipulation between edges in order to mitigate that risk. Hence deliberate removals and additions are being done in order to change the shape of the graph and increase the difficulty to link nodes with the same location of the previous snapshot which is called graph perturbation [8].



Above, the simple time-varying graphs are represented. As it can be seen, the edge between (A, B) and (D, E) is gone in $t+1$ time and a new node F with new edges to B and F is created. It is important to note that the deliberate addition and removal of nodes cannot be distinguished from the changes that have actually occurred in the real world.

The comparison between time-varying graphs is done in subsequent releases, in a way that the newly released snapshot of the graph is taken in time $t + 1$ where the previous one is taken in time t. Hence, the graph representation of graph is $G^{t+1}(V, E)$ and the previously released graph is $G^{t}(V, E)$.

Representation of Data and Quantification of Changes: Each attribute in the columns can be represented as a vector of sub-attributes such as

$$v = (s_1, s_2, s_3, ..., s_k) \tag{1}$$

and the difference between the two attributes in the time-varying graphs would be simply

$$v^{t,t+1} = \sqrt{\sum_{n=0}^{k} (s_n^{t+1} - s_n^{t})^2} \tag{2}$$

by the Euclidean distance.

The differences between each node in the graph are represented as follows:

$$\delta_i = \sum_{i=0}^{|V|} v_i^{t,t+1} \tag{3}$$

The goal of having this equation is to calculate the square root distance (Euclidean distance) between each identifier of the current and the previous snapshot as given in Equation 2. The attributes are domain-specific and the parameters are set accordingly. Those attributes should not be published to the public. Similar functionality is also used for machine learning algorithms such as kNN or clustering purposes as unsupervised techniques [9].

The difference between edges in time-varying graphs is rather simple and can be represented as the sum of all absolute differences in each node's edges.

$$\sum_{i,j \& i \neq j} \| e_{i,j}^{t+1} - e_{i,j}^{t} \|, e \in E, i, j \in \{0, |E|\} \tag{4}$$

In order to reduce the complexity and prevent the research from diverging, we did not consider the differences between edges in time-varying graphs as temporal data.

Temporal Data: Temporal data refers to data that changes over time [10]. This change may include attributes of specific entities which cause privacy risks. That is, attackers can benefit from temporal data change to de-anonymize and identify individuals. The quantification of temporal data changes causing temporal de-anonymization can be calculated by Equation 1.

Significant changes increase the risk of deanonymization. However, it is also important to note that such differences might have significant contributions to literature. Hence, it is important to provide this data while ensuring the anonymity of the user.

Normal Data: Time-varying graphs are not containing only temporal data but also normal data which is the change that is not causing privacy risks for de-anonymization. Such changes should be small which can be determined by Equation 1 if it is less than some threshold.

Attacker Model: Since there are tremendously many different types of adversaries and techniques to attack users' privacy, we had to limit our constraints in the attacker model. The first assumption is that attacker has access to the individuals' temporal data in real life. However, the attacker does not know which nodes the identity corresponds to. The second assumption is that attacker is only able to investigate two subsequent time-variant graphs to achieve their goal even if they have access to previous graphs. The third assumption is that no correlation can be made with other identifiers besides the temporal data to the real identity.

The goal of the attacker is to link the real identity to the node in the graph through temporal data. Hence, we had to assume that the attacker has access to such temporal data and has the knowledge to whom it belongs as it is stated in the first assumption.

The second assumption has been made since all the anonymization techniques that are being used are taking into consideration two subsequent time-varying graphs. If there is high coherence between graphs differing from more than one snapshot interval, then it will automatically make all the techniques insufficient for sustaining an individual's privacy.

We are investigating the de-anonymization risks using temporal data, hence, if the identifier can be linked to the real identity, then it must be temporal data. Otherwise, social media companies are already pre-processing graphs and mitigating the de-anonymization risk for normal changes in data assuming that their algorithm is efficient. If data is not linkable to the identity, then it is not important. These interpretations have enabled us to make the last assumption.

B. Structure of Social Media Sites

1) Overview: Social Media Sites such as Facebook, Snapchat, etc. share their data anonymously in graph format [11]. The representation is a graph, however, the connections between edges and the properties of nodes are shared in two different text files. The edges are printed to the text file in the following format $edge1, edge2$ for each line. However, the node representation can be either like $nodeID, attr1, attr2, ..., attrk$ or basically in JSON format.

2) Simulated Graphs: When we construct our simulated graphs, we used JSON format but for some demonstration purposes, we used tables where some of the nodes are placed into rows where the columns form the attributes.

Temporal data is selected from the clusters randomly created at the beginning. In order to make the simulation close to a real one, we did not put too much difference in the nodes' attributes and edges except for some small changes and temporal data in the next snapshot iteration.

Since our research is focused on temporal data, we did not give any credit for the edges and the changes on them. Significant reduction or increase in edges and shapes can also become temporal data. However, for this paper, we only focused on nodes' attributes' changes.

C. Anonymization Techniques

In this section, we investigate the popular k-anonymity algorithms being used for the anonymization of graphs. Also, we demonstrated and proved their insufficiency regarding the anonymization of temporal data which leads to the de-anonymization risks. Their techniques are described below.

All the techniques analyzed below are related to k-Anonymity. There are also other concepts such as l-diversity and t-closeness, but we limited ourselves to widely used techniques.

For all methods, it is important to keep in mind that the techniques will be applied only if there is temporal data in time-varying graphs. In other words, if the adversary knows the significant changes in attributes, then they will be looking for such changes in subsequent graphs. Hence, the anonymization techniques will be applied for temporal data in graph $G^{t+1}(V, E)$. Equation 5 shows the equation that determines the change whether it is temporal or not where k is the index for the sub-attribute of attributes in vector described in Equation 1.

$$\lambda \leq \sqrt{\sum_{k=0}^{|v|} (v^t(k) - v^{t+1}(k))^2} \quad v \in V \tag{5}$$

In order to determine whether the changes are temporal, we can set the threshold as a constraint for Euclidean Distance.

1) Generalization: The first method we investigated that aims to achieve k-anonymity is a generalization which is reducing an attribute's specificity if there are cases where the attribute reveals identifying information. This can be achieved by either reducing the information presented by an attribute or even completely omitting it. For example, in data sets where the zip code and the age attributes of a row of data present enough information for identification the zip code may be generalized to only reveal the municipality (068XX as opposed to 06830) and/or the specific age may be generalized to an age group (18 to 15-20). By this reduction we can satisfy k-anonymity however due to each attribute corresponding to a higher scope now some utility is naturally lost.

There are two main concerns with this approach. The first is the loss of information and thus utility. Generalization-based k-anonymization runs the risk of over-generalizing as quasi-identifiers, which are the set of identifiers that can uniquely identify an individual in combination [12], provide crucial information regarding the sensitive information of the dataset. Therefore, considering both the temporal changes in the database and possibly including other bits of data along with the quasi-identifiers, some attributes may become borderline obsolete. This results in both discarded records and attributes where the information loss can go all the way up to 30% for high Ks [13]. The second concern is the time and space complexity of the techniques developed utilizing generalization. Many algorithms proposed for this technique follows methods similar to; checking if k-anonymity is satisfied, selecting the possibility with the minimal distortions out of the possible generalizations, repeating if k-anonymity is not satisfied [14] [15]. Naturally, an exhaustive search over all the possible generalizations results in infeasible run-times over large-sized tables of data like that of social networks.

2) Data Masking: Data masking solution is designed to re-initialize data, which means that data remains related to the real information but data no longer has practical usage. In other words, it is just noisy data rather than information.

The masking in temporal data will prevent attackers to link the node to the real identity since the temporal data will be masked. However, even though it will help to reduce linkability risk, the utility will be reduced as the attributes in nodes will not make any sense to the researchers [16].

3) Data Swapping/Shuffling: Our third method that can be used for temporal anonymization is data swapping, also known as data shuffling. Data shuffling can be described as mixing and replacing existing data with values in different rows while preserving their own values [17]. This mixing operation is the process of vertically mixing the existing values in the dataset in a random way. For example, shuffling the values of individuals in the column of salaries in a table containing the data of employees in a company serves to hide the private salaries of individuals and also provides anonymity without harming the usefulness of the table. On the other hand, it does not create a problem in calculating the total or average salary values in the company. Data Shuffling is a common process used to hide the relationship of data that may be sensitive and private while preserving aggregate values [18]. Since the general structure of the table is not deteriorated by this technique, the statistical values of the tables can be used safely for testing and training purposes. However, besides the statistical usages, the other utility is damages lot.

This method can be used in various ways such as *Random Shuffling*, *Designating Groups*, *Designating Partitions*.

Random Shuffling is the process of mixing and replacing values that may be sensitive without a specific rule with different values on the vertical column. The random shuffling without considering the other attributes will kill the utility of the data. Hence, it is proven to be an inefficient algorithm since it does not satisfy the constraints on utility.

In the Designating Groups operation, related values in other columns in the same row are shuffled with the values in different rows in the group that corresponds to these values. Hence, all the attributes will remain in conjunction but in different nodes. Shuffling the attributes as a whole with other nodes will destroy the graph property as the nodes will be irrelevant in time-varying graphs.

In the Designating Partitions process, the columns containing the data are mixed in the appropriate sections and are not associated with the values in different sections. To simplify, temporal data will be replaced with other data in the same column with the row that has the same attribute value except for the column where temporal data belongs. If the position and the properties are the same except for the temporal data, then the utility might not suffer from shuffling. However, the algorithm will fail when there are no similar rows in the graph which is also the main problem of k-anonymity.

III. Proposed Method

Our constraints for the anonymization of temporal data problem are preserving utility and protecting the anonymity of the users. Hence, we introduced a new algorithm for temporal data anonymization to satisfy both constraints.

Our algorithm is run for time-varying graphs as it is the scope of this paper. For subsequent time-varying graphs, it looks for changes on each attribute and calculates the distance by the Euclidean distance formula, as it is stated in *line 7*. If the differences are quite large than the predefined σ value, which is specific to the domain, then it interprets it as temporal data and proceeds accordingly.

We included Binomial probability randomness to either not reflect new data or to replace it with the closest substitutes as it is stated in *line 8* [19]. In the cases that the binomial value is 1, then one of the closest substitutes will be replaced with the attribute in $node^{t+1}$. In the cases the binomial value is 0, then the change will not be reflected the old value will remain and the new value will not be reflected. Eventually, the temporal value will be reflected in the following graphs. The proof of that is given in the following section.

For selecting the substitutes, we used the $vClosest$ algorithm which finds the closest substitutes by Euclidean distance. Both σ and sub-attributes of the attributes are domain-specific and should be pre-defined. In order to protect the operability of the algorithm, the tables should not be published. Otherwise, the adversary can find the closest attributes by running Euclidean distances and searching for them accordingly.

For each attribute in the domain, the $vClosest$ algorithm calculates Euclidean distance with temporal data, and if it is less than σ, it puts it into the result array. Hence, all items in the return list of the $vClosest$ algorithm are guaranteed to be less than σ which makes all of them close substitutes for small σ.

The important point of the $vClosest$ algorithm that needs attention is that it is also selecting the temporal data from the domain as its distance with itself is 0 for $\sigma > 0$. This trivial case has been put intentionally to guarantee that the list will contain at least one element. In some cases when σ is too small or there is not a sufficient amount of attributes in the domain, then the list will not contain any substitutes resulting in no attribute to be put into the node corresponding to the temporal data besides itself. Hence, it is equally crucial to have optimal σ and a large domain.

For nodes that are deleted from the graph or added at the instance between t and $t+1$, the algorithm does not takes it into the consideration as the total Euclidean distances for each sub-attribute will be large as

$$\delta = \sqrt{\sum_{n=0}^{k} (s_n^{t+1} - 0)^2} = \sqrt{\sum_{n=0}^{k} (s_n^{t+1})^2} \tag{6}$$

in added node making it temporal most of the time unless the sum of the distances for all attributes is not very small. A similar relationship occurs in the deletion of node and the distance is the same with addition as

$$\delta = \sqrt{\sum_{n=0}^{k} (0 - s_n^{t})^2} = \sqrt{\sum_{n=0}^{k} (s_n^{t})^2} \tag{7}$$

Even though the changes can be interpreted as temporal in deletion or addition of nodes, this can be anonymized using graph perturbation methods as deciding to add it into the next instance or leave it for the next iterations. Also, the adversary cannot be sure of the node's identity since it will not know whether the user creates an account at that instance or not. In case of deletion, the adversary will not be able to link identities to the nodes as there will not be any nodes.

Lastly, our algorithm removes the personal identifiers and assigns some number as an ID for each node.

Algorithm 1 Anonynimization algorithm for temporal data

```
function AnonymizeTemporalData($nodes^t$, $nodes^{t+1}$, $domains$, $k$)
    $i \leftarrow 1$;
    $j \leftarrow 1$;
    for $i \leq nodes^{t+1}.count$ do;
        if $nodes^t \quad exists$ then
            for $j \leq nodes^{t+1}[i].columnCount$ do
                if $euclideanDistance(nodes^t[i][j], nodes^{t+1}[i][j]) > \sigma_j$ then    ▷ $\sigma_j s$ is determined specific to the domain
                    $binRandom \leftarrow randomBinomial()$
                    if $binRandom = 1$ then
                        $(m_1, m_2, m_3, ..., m_k) \leftarrow vClosest(nodes^{t+1}[i][j], domains[j], \sigma_j)$
                        $nodes^{t+1}[i][j] \leftarrow random(m_1, m_2, m_3, ..., m_k))$
                    else
                        $nodes^{t+1}[i][j] \leftarrow nodes^t[i][j]$
                    end if
                end if
                $j \leftarrow j + 1$
            end for
        end if
        $i \leftarrow i + 1$
    end for
end function
```

Algorithm 2 Finding v Closest attributes

```
function vClosest($attribute$, $domain$, $\sigma$)
    $i \leftarrow 1$
    $distances \leftarrow []$
    for $i \leq domain.count$ do
        $distance \leftarrow euclideanDistance(attribute, domain[i])$
        if $distance < \sigma$ then
            $distances[i] \leftarrow (i, euclideanDistance(attribute, domain[i]))$
        end if
        $i \leftarrow i + 1$
    end for
    return $distances.firstColumn$
end function
```

A. *Proofs*

Claim: It is guaranteed that temporal data value will appear at the time-varying graphs at some instance.

Proof: Assume that the binomial value is 1 for the temporal data in between graphs between t and $t+1$ instances. Also, assume that the old value for temporal data is a^t, the temporal value is a^{t+1} and the closest substitute is b^{t+1}. In case $a^{t+1} = b^{t+1}$, then the temporal data will be reflected on the graph immediately. If $a^{t+1} \neq b^{t+1}$ and b^{t+1} is one of the other substitutes, and if there are no temporal data changes between a^{t+1} and a^{t+2}, the substitute for the temporal data at instance $t+1$, which is b^{t+1}, will be replaced with a^{t+2}. In case the temporal changes occur subsequently, and each attribute is replaced with b^{t+i} instead of a^{t+i}, then at some point, $t+k$, the a^{t+k} will be reflected. With the same assumption, the probability of the temporal data not being reflected at time instance $t+k$ for consistent temporal changes from t to $t+k$ is

$$p = \left[\frac{v-1}{2v}\right]^k \tag{8}$$

as the probability of binomial is $1/2$ and the possibility of not choosing a^{t+i} is $(v-1)/v$ for each time instance with an assumption that the substitute set size for each temporal data is v.

As k diverges to ∞

$$p = \lim_{k \to \infty} \left[\frac{v-1}{2v}\right]^k = 0 \tag{9}$$

Hence, it is not possible to not reflect the temporal data in the long run as the probability decreases substantially in each iteration. The only possible cases for not being reflected at all are either when a node is deleted before or when there is another change that occurs in the attribute at some time instance.

On the other hand, if binomial is 0, then no changes with be reflected in a^{t+1} and if it is also the case for $t+2$, then

a^{t+1} with be put into a^{t+2} and it will keep going to reflect the previous values unless binomial is 1. The probability converges to 0 as the iteration number increases. Hence, at some point, the binomial will be 1, which will result in the outcome that is described in above.

Claim: Utility of the graph will be preserved in long run.

Proof: The Euclidean distances are the smallest ones in the clusters that the attributes belong to. Hence, the attributes are related closer to each other than random attributes. If the set of substitutes is S

$$\sigma > v, \forall v \in S \tag{10}$$

and

$$\sigma > \sqrt{\sum_{n=0}^{k} (s_n^{t+1} - s_n^t)^2} \tag{11}$$

where σ is the constraint for the distance in the domain.

As a result, the utility will be preserved as the substitutes are close to the temporal data. In case the binomial is 0, then at some point, it will reflect the substitute or/then temporal data in the next iterations proved in the first claim.

Claim: The risk of de-anonymization through temporal data is reduced significantly.

Proof: In case the binomial is 0, then the change will not be reflected, hence, there will be no changes to link the users to the node through temporal data. On the other hand, if the binomial is 1, then the closest substitute will be selected for $t+1$. Since the attacker cannot correlate the temporal data with its substitutes, the anonymity will be preserved. Another case can occur only when the temporal data itself is chosen as a substitute from the set while the binomial is 1. In that case, assume that the probability of linking a node to its user is p for the adversary. Then, the new possibility after anonymization with our algorithm will be

$$q = p * \frac{1}{2} * \frac{1}{v} = \frac{p}{2v} \tag{12}$$

and

$$p > \frac{p}{2v} \tag{13}$$

meaning that the probability of de-anonymization risk through temporal data is reduced significantly. As it is seen in the comparison, the probability is highly dependent on v, the size of the close substitutes set.

The algorithm is providing utility while preserving the anonymity of the users with respect to their temporal data derived from the proofs given above.

B. Limitations

In this section, we provided limitations of our solution for temporal anonymization.

1) Quantification of Attributes: It is very important to be able to quantify attributes and also come up with a solution that will enable the algorithm to calculate Euclidean distance correctly for close substitutes. In some cases, the quantification can not be done due to the limited number of attributes in the domain such as gender. In this case, the algorithm will not be able to run efficiently.

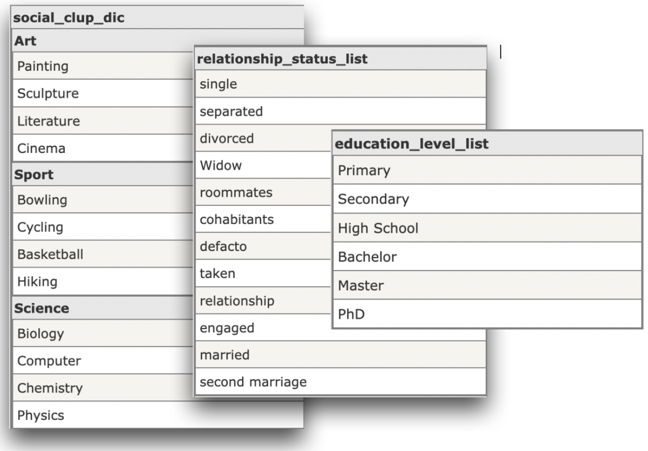
2) Same Temporal Changes in Multiple Nodes: Our algorithm focuses on temporal data, hence, if there are many nodes having the same temporal data such that $v_i^t - v_k^{t+1}$ where v_k^{t+1} is temporal data and a total number of $v_i^t - v_k^{t+1}$ changes is greater than 1 at instance $t+1$, then the algorithm will try to anonymize each of them. However, such large changes can provide k-Anonymity which might not require any additional anonymization.

IV. Experiment

In this section, we experimented with the efficiency and accuracy of our method for the anonymization of temporal data which is described in the previous section. Our method is applied to time-varying graphs as we discussed. It searches for changes in each attribute for subsequent time-varying graphs and employs the Euclidean distance method to calculate distance. It interprets the changes as temporal data only if they are above a predefined threshold that is specific to the domain. We utilize binomial probability randomness to either not reflect new data or to replace it with the most similar substitutes. If the binomial value is 1, the attribute in the node will be replaced by one of the closest substitutes. The change will not be reflected, if the binomial value is 0, therefore the old value will remain and the new value won't be reflected. In this section, we mainly talk about our experiment for testing our proposed method and our experiment setup.

A. Simulated Data

For testing our method, our first job was to create a graph and temporal dataset. For that purpose, we simulate simple social network data that includes users' personally identifying information (PII), and quasi-identifiers. In our data as a PII, we hold the names of the users and as a quasi-identifier, we hold users' social club list, education level, and relationship status. For the sake of simplicity, we consider temporal data of the form D[1], D[2], and D[3] where each dataset corresponds to the snapshot containing the records collected at the time instance i where $i \in \{1, 2, 3\}$. For each instance, we take 3 snapshots from 2018, 2020, and 2022. The domain of each attribute in a dataset should be carefully defined because it can affect the accuracy and value of the data. Therefore, before discussing our experiment, we would also like to discuss our attributes' domain. For each attribute in our domain, we create a variety of feature types. For instance, since each user has a list of social clubs, we construct three subdomains for that attribute domain: sports clubs, art clubs, and scientific clubs. Each user is permitted to participate in up to three social clubs. We present our attribute domain in Figure 1.



social_clup_dic
Art
Painting
Sculpture
Literature
Cinema
Sport
Bowling
Cycling
Basketball
Hiking
Science
Biology
Computer
Chemistry
Physics

relationship_status_list
single
separated
divorced
Widow
roommates
cohabitants
defacto
taken
relationship
engaged
married
second marriage

education_level_list
Primary
Secondary
High School
Bachelor
Master
PhD

Fig. 1. Attribute Domain

B. Experiment Setup

For building our data, we write a simple python code that randomly assigns a value for each attribute of the user from the defined attribute domain.

We made three copies of the same dataset after creating users' data and added the database identifier and time value for generating the data snapshots. Since we replicated the same data, in the beginning, there were no temporal changes. We have constructed a change set and integrated it into our datasets in order to create temporal changes. All the changes that we applied can be found in Table I. To make the simulation close to a real one, we did not put too much difference in data attributes.

The assignment of the value to each attribute is done according to the closest attribute intuitionally. The values that are assigned for attributes in domains are given in Appendix A. The identities of the users that are assigned to nodes in the graph are given in Appendix B. Lastly, the log file that shows the detection of temporal data and their changes with closest substitutes is given in Appendix C.

We replaced the user's name with an identity number after constructing the temporal dataset in order to eliminate PII. Then, in order to use each domain element in our experiment, we provide it with a specific value in accordance with their closeness. For instance, basketball and hiking both belong to sports clubs, and their respective values are 13 and 14. However, since the chemistry club is a subset of the science club, its value is 22, which is distant from the sports club attributes domain values.

The distance constraints for domains are chosen as a social club to 2, relationship status to 2, and education level to 4.

V. Results and Discussion

In Table II, the red color shows the temporal data being replaced with substitutes. The blue are the ones that are being detected to be temporal but not changed due to the binomial value. Lastly, the green ones represent the normal changes which do not satisfy temporal constraints.

TABLE I
Temporal Data Changes

Name	2018	2020	2022
Nylah Walker	single	married	married
Cheyenne Coleman	Literature	–	–
Tucker Gates	High School	High School	Master
Rylie Barton	–	Painting	Painting
Ansley Park	Primary	Primary	Secondary
Madeleine Hopkins	married	de facto	divorced
Omari Fitzgerald	–	–	Basketball
Jaquan Hopkins	Secondary	Bachelor	Bachelor
Bobby George	Basketball	–	–
Gaige Christian	married	married	divorced
Rylee Green	Bachelor	Bachelor	Master
Eva Parrish	–	Cycling	Cycling

TABLE II
Anonymized Temporal Data Changes

Name	2018	2020	2022
Nylah Walker	single	second marriage	married
Cheyenne Coleman	Literature	Literature	–
Tucker Gates	High School	High School	Bachelor
Rylie Barton	–	–	Painting
Ansley Park	Primary	Primary	Secondary
Madeleine Hopkins	married	married	de facto
Omari Fitzgerald	–	–	–
Jaquan Hopkins	Secondary	Secondary	Bachelor
Bobby George	Basketball	Basketball	–
Gaige Christian	married	married	divorced
Rylee Green	Bachelor	Bachelor	Master
Eva Parrish	–	–	Cycling

The summary of the data for both temporal and normal is given in Table III. The total number of normal data is 2 and temporal is 11. As it is also stated in the table, there are no false hits that prove our algorithm's accuracy claim. This outcome is resulted due to the accurate predefined constraints for each domain.

The statistics for binomial randomness and the changes with substitutes or itself are given in Table IV. The probability for binomial randomness is set to 0.5 but it is observed that the ratio of numbers of 0 to 1 is $8/11$ which is highly greater than 0.5. This can be explained by the size of the changes as it is very small. With a large number of temporal changes in graphs, the binomial ratio will diverge to 0.5 as it is one of its properties.

TABLE III
Temporal Data Detection

Data	Correct	Incorrect
Normal	2	0
Temporal	11	0

TABLE IV
Temporal Data Changes Results

	Binomial	Substitute
No	8	1
Yes	3	2

Algorithm 3 Creating the Dataset

```
function BUILDDATA($name\_list$, $social\_clubs$, $education\_level$, $relationship\_statuses$)
    $graph \leftarrow \{\}$
    $identities \leftarrow \{\}$
    $i \leftarrow 1$ // assuming that the name and surname is unique
    for $userinname\_list$ do
        $clubs \leftarrow []$
        for $x \leq random.range(0, 4)$ do // select club size for each user
            $club \leftarrow social\_clubs.getRandom()$
            if $club \notin clubs$ then
                $clubs.append(club)$
            end if
        end for
        $education \leftarrow education\_level.getRandom()$
        $relation \leftarrow relationship\_statuses.getRandom()$
        $user \leftarrow createUser(clubs, education, relation)$
        $graph.append(user)$
        $identities.append(i, name)$
        $i \leftarrow i + 1$
    end for
end function
```

On the other hand, 1 data remained the same according to Table IV and 2 had changed to substitutes. As it is explained in Section III, the $vClosest$ function adds the data itself to the closest substitutes, hence, there is a possibility for temporal data not being anonymized at all.

Figure 2 shows the total value of divergence from real attributes according to the different thresholds being set for temporal constraint. The blue line shows the average distances between the first snapshot and the second snapshot. For simplicity, all threshold is set to be the same but it can not be the case. The average distances for each threshold value is set after running 100 iterations. As can be seen in the table, there are no significant changes in averages. This outcome is resulting from the 2 facts which are each attribute is changed by either its close substitutes or remained the same with the previous snapshot's attribute or directly reflects temporal data, and the threshold is relatively small. For a large threshold, it will diverge significantly. A relatively small increase can be observed in both snapshots 1-2 and 2-3 from thresholds 4 to 7.

For threshold 0, it is observed that the average distance is not 0. This output is obtained because of the binomial randomness if it is 0 then it will not reflect the temporal data which results in the such distance.

TABLE V
DE-ANONYMIZED OF NODES

	de-anonymization
Success	0
Fail	11

As it is shown in Table V, we could not de-anonymize even a single user successfully. We were able to correlate the changes with nodes but they were false correlations. Even though we had 1 temporal change not being anonymized, which is Rylee Green, we could not link to the user as there were multiple instances of married-divorced in the graphs at instances 2 and 3.

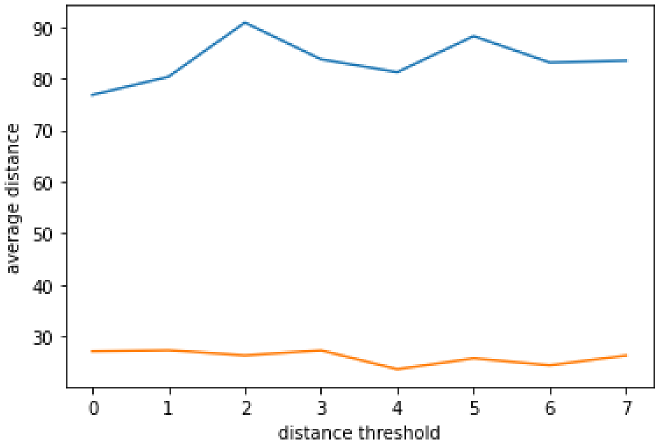


Fig. 2. Avg. distances for snapshots vs thresholds (blue: 1-2, orange: 2-3)

VI. CONCLUSION

In our paper, we investigated temporal data changes and their effect on the de-anonymization of users in time-varying social network graphs. We were able to quantify temporal data and perform a risk analysis of temporal de-anonymization. Using the simulated data, we successfully anonymized temporal data while preserving the utility of the graphs, and the algorithm is proven to be successful on similar types of datasets in real-world examples. Our algorithm is not interested in nodes and normal data changes however, it is designed to be integrative into any other algorithms that will be used to boost the anonymity of the nodes.

The attack for de-anonymization of nodes with temporal data remains simple. However, this attack can be developed with additional auxiliary information, which we did not do. At this point, additional anonymization techniques to prevent such attacks can be developed on top of our proposal. Nonetheless, our algorithm is a practical and efficient one which makes it a good starting point for further investigation of temporal data.

Acknowledgment

The authors would like to express their deep gratitude to Asst. Prof. Dr. Erman Ayday, our instructor, for his patient guidance, enthusiastic encouragement, and useful critiques of this research work and its presentations.

References

[1] A. Esteve, "The business of personal data: Google, Facebook, and privacy issues in the EU and the USA," *International Data Privacy Law*, vol. 7, pp. 36–47, 03 2017.

[2] V. Kumar and R. Mirchandani, "Increasing the roi of social media marketing," *MIT Sloan Management Review*, vol. 54, pp. 55–61, Fall 2012. Copyright - Copyright © Massachusetts Institute of Technology, 2012. All rights reserved; Document feature - Tables; ; Last updated - 2022-12-16; CODEN - SMRVAO.

[3] J. Isaak and M. J. Hanna, "User data privacy: Facebook, cambridge analytica, and privacy protection," *Computer*, vol. 51, no. 8, pp. 56–59, 2018.

[4] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow, "The anatomy of the facebook social graph," *CoRR*, vol. abs/1111.4503, 2011.

[5] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel, "A practical attack to de-anonymize social network users," in *2010 IEEE Symposium on Security and Privacy*, pp. 223–238, 2010.

[6] A. Halimi and E. Ayday, "Efficient quantification of profile matching risk in social networks," *CoRR*, vol. abs/2009.03698, 2020.

[7] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro, "Time-varying graphs and dynamic networks," pp. 346–359, 2011.

[8] V. Torra and J. Salas, "Graph perturbation as noise graph addition: A new perspective for graph anonymization," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology* (C. Pérez-Solà, G. Navarro-Arribas, A. Biryukov, and J. Garcia-Alfaro, eds.), (Cham), pp. 121–137, Springer International Publishing, 2019.

[9] I. Dokmanic, R. Parhizkar, J. Ranieri, and M. Vetterli, "Euclidean distance matrices: Essential theory, algorithms, and applications," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 12–30, 2015.

[10] C. S. Jensen and R. T. Snodgrass, "Temporal data models," pp. 2952–2957, 2009.

[11] N. A. Shozi and J. Mtsweni, "Big data privacy in social media sites," pp. 1–6, 2017.

[12] L. Sweeney, "K-anonymity: A model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, p. 557–570, oct 2002.

[13] E. Armengol and V. Torra, "Generalization-based k-anonymization," in *MDAI*, 2015.

[14] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, p. 571–588, oct 2002.

[15] L. Rossi, M. Musolesi, and A. Torsello, "On the k-anonymization of time-varying and multi-layer social graphs," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 9, pp. 377–386, Aug. 2021.

[16] G. Sarada, N. Abitha, G. Manikandan, and N. Sairam, "A few new approaches for data masking," pp. 1–4, 2015.

[17] O. O. Ajayi, "Application of data masking in achieving information privacy," *IOSR Journal of Engineering*, vol. 4, pp. 13–21, Feb. 2014.

[18] K. Muralidhar and R. Sarathy, "Data shuffling—a new masking approach for numerical data," *Management Science*, vol. 52, pp. 658–670, May 2006.

[19] F. Mosteller and J. W. Tukey, "The uses and usefulness of binomial probability paper," *Journal of the American Statistical Association*, vol. 44, pp. 174–212, June 1949.

APPENDIX A
DOMAINS OF THE SIMULATED DATA

domains.json

```
{
        "social_club": {
                "None": [1],
                "Bowling": [11],
                "Cycling": [12],
                "Basketball": [13],
                "Hiking": [14],
                "Biology": [21],
                "Chemistry": [22],
                "Physics": [23],
                "Computer": [24],
                "Painting": [31],
                "Sculpture": [32],
                "Literature": [33],
                "Cinema": [34]
        },
        "relationship_status": {
                "single": [1],
                "separated": [2],
                "divorced": [3],
                "Widow": [4],
                "roommates": [10],
                "cohabitants": [11],
                "de facto": [12],
                "taken": [14],
                "relationship": [15],
                "engaged": [17],
                "married": [18],
                "second marriage": [19]
        },
        "education_level": {
                "Primary": [1],
                "Secondary": [5],
                "High School": [9],
                "Bachelor": [13],
                "Master": [17],
                "PhD": [19]
        }
}
```

APPENDIX B
REAL NAMES AND ID OF USERS IN SIMULATED DATA

identities.json

```
{
    "Nylah Walker": 1,
    "Cheyenne Coleman": 2,
    "Tucker Gates": 3,
    "Rylie Barton": 4,
    "Taylor Wells": 5,
    "Ansley Park": 6,
    "Sabrina Bonilla": 7,
    "Paul Hunter": 8,
    "Andres Pittman": 9,
    "Madeleine Hopkins": 10,
    "Omari Fitzgerald": 11,
    "Corey Lambert": 12,
    "Franklin Esparza": 13,
    "Kara Watkins": 14,
    "Brennen Montgomery": 15,
    "Jaquan Hopkins": 16,
    "Bobby George": 17,
    "Angel Woods": 18,
    "Baylee Bolton": 19,
    "Yaretzi Sweeney": 20,
    "Darrell Gonzalez": 21,
    "Clara Hickman": 22,
    "Malachi JuarezCaitlin Lang": 23,
    "Adeline Peters": 24,
    "Braelyn Bowman": 25,
    "Abby Decker": 26,
    "Sammy Pugh": 27,
    "Gaige Christian": 28,
    "Joanna Moyer": 29,
    "Kailee Lawrence": 30,
    "Lilly Cunningham": 31,
    "Kareem Williamson": 32,
    "Jaelynn Harris": 33,
    "Eva Parrish": 34,
    "Rylee Green": 35
}
```

APPENDIX C
THE LOG FILE OF THE ALGORITHM DISPLAYING LOG OF CHANGES

log.txt

```
node id: 1
temporal t1: single , t2: second marriage , new_value: second marriage ,
closestSubstitutes: [(1.0, ’engaged’), (0.0, ’married’), (1.0, ’second marriage’)]

node id: 2
temporal Literature not removed by binomial

node id: 4
temporal Painting not added by binomial

node id: 10
temporal t1: married , t2: married , not added by binomial

node id: 16
temporal t1: Secondary , t2: Secondary , not added by binomial

node id: 17
temporal Basketball not removed by binomial

node id: 34
temporal Cycling not added by binomial

node id: 3
temporal t2: High School , t3: Bachelor , new_value: Bachelor ,
closestSubstitutes: [(4.0, ’Bachelor’), (0.0, ’Master’), (2.0, ’PhD’)]

node id: 6
normal t2: Primary , t3: Secondary

node id: 10
temporal t2: de facto , t3: de facto , not added by binomial

node id: 11
temporal Basketball not added by binomial

node id: 28
temporal t2: married , t3: divorced , new_value: divorced ,
closestSubstitutes: [(2.0, ’single’), (1.0, ’separated’), (0.0, ’divorced’), (1.0, ’Widow’)]

node id: 35
normal t2: Bachelor , t3: Master
```