Privacy Preservation of Social Network Users Against Attribute Inference Attacks via Malicious Data Mining

Khondker Jahid Reza[1], Md Zahidul Islam[1] and Vladimir Estivill-Castro[2]
[1]*School of Computing and Mathematics, Charles Sturt University,*
Panorama Avenue, Bathurst 2795, NSW, Australia
[2]*Departament de Tecnologies de la Informació i les Comunicacions, Universitat Pompeu Fabra,*
Roc Boronat, 138, 08018 Barcelona, Spain

Keywords: Attribute Inference, Data Mining, Privacy Protection Technique.

Abstract: Online social networks (OSNs) are currently a popular platform for social interactions among people. Usually, OSN users upload various contents including personal information on their profiles. The ability to infer users' hidden information or information that has not been even uploaded (i.e. private/sensitive information) by an unauthorised agent is commonly known as *attribute inference problem*. In this paper, we propose *3LP+*, a privacy-preserving technique, to protect users' sensitive information leakage. We apply *3LP+* on a synthetically generated OSN data set and demonstrate the superiority of *3LP+* over an existing privacy-preserving technique.

1 INTRODUCTION

Data mining of users' information on Online Social Networks (OSNs) can reveal individuals' private information. News pieces, reported by the Boston Globe (Johnson, 2009), reveals that users' sexual orientation can be correctly predicted by accessing their non-sensitive information. Hence, the ability of an intruder to infer users' sensitive information is known as *the attribute inference problem*.

A technique (*PrivNB*), based on the Naïve Bayes classifier, can protect a sensitive attribute of users by suppressing attribute values identified as predictors and deleting some friendship links (Heatherly et al., 2013). Another technique called *3LP* (Reza et al., 2017a) uses a decision forest algorithm to enhance users' privacy against *the attribute inference problem*. *3LP* in its first layer suggests to a victim user which predictor attribute values to suppress from the user's profile. If the sensitive information of the user is still unprotected even after the suppression of predictor attributes, then *3LP* suggests to hide some existing friends from the user's friend list in its *Layer 2* and add new friends in *Layer 3*.

In this study, we propose *3LP+* which is an extension of the existing *3LP* algorithm (Reza et al., 2017a). *3LP* assumes the existence of a single sensitive attribute such as the "*Political View*" while in reality a user is likely to have multiple sensitive attributes such as the "*Political View*" and "*Sexual Orientation*". To protect the privacy of multiple sensitive attributes *3LP* could be applied multiple times, but every run of *3LP* would be isolated/independent. As a result, they can be counterproductive in the sense that one run (say to protect the "*Political View*") might suggest hiding a friendship information with another user while a subsequent run (say to protect "*Sexual Orientation*") might suggest disclosing the same friendship information resulting in the loss of protection of "*Political View*".

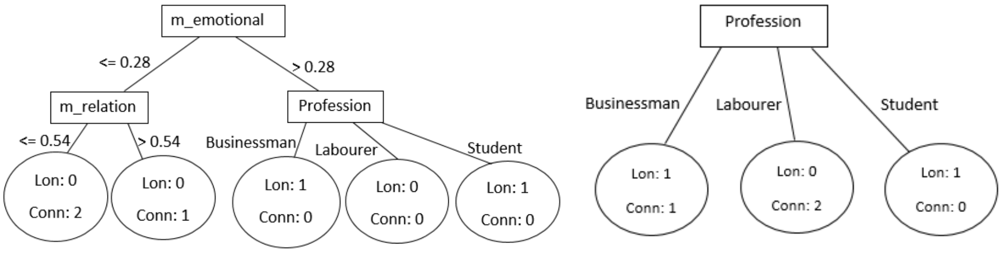
3LP+ aims to provide privacy for multiple sensitive attributes through a co-ordinated approach as opposed to the isolated approach. It uses a matrix to store the history of any friendship being hidden or new frinedship being created during a run to avoid a conflicting suggestion in a subsequent run. For example, if the t-th run suggests hiding a friendship of the victim user with another user (and the victim user actions on the suggestion), then the matrix stores that information so that a subsequent run does not suggest the victim user creating the friendship with the same user.

The rest of this paper is organized as follows. In Section 2, we describe the privacy attack model considered in this study. Section 3 presents the privacy-preserving technique *3LP+*. We describe our experi-

Reza, K., Islam, M. and Estivill-Castro, V.
Privacy Preservation of Social Network Users Against Attribute Inference Attacks via Malicious Data Mining.
DOI: 10.5220/0007390404120420
In *Proceedings of the 5th International Conference on Information Systems Security and Privacy (ICISSP 2019)*, pages 412-420
ISBN: 978-989-758-359-9

Table 1: A sample data set.

User	Relationship status	m_relation	Profession	m_profession	m_emotional	Class Attribute
a	Married	0	Businessman	0	0.56	Lonely (Lon)
b	Widow	0	Businessman	0	0	Connected (Conn)
c	Single	0.56	Labourer	0	0	Connected
d	Married	0.51	Student	0.56	0.56	Lonely
e	Single	0.51	Labourer	0	0	Connected
u	Widow	0.51	Student	0.51	0	?



(a) Tree-1

(b) Tree-2

Figure 1: Decision trees build on the sample data set given in Table 1.

mental set up in Section 4 and the experimental results in Section 5. Section 6 gives a concluding remark.

2 PRIVACY ATTACK SCENARIO

An attacker M can follow any approach to invade users' privacy. We consider a data mining approach, in this study, assuming that, M prepares a data set by analyzing an OSN of N users, each with A attributes and friendship information. We name an attribute, and it's corresponding value together as an attribute-value pair. Here each attribute value ($A_n = v$) is considered as a distinct binary attribute. In a Social attribute network (*SAN*) model, both users and their attribute-values are modeled as vertices.

The attacker can take advantage of a metric function as shown in Equation 1 (Adamic and Adar, 2003), to incorporate the friendship information into D.

$$m(u,A_n = v) = \sum_{t\in\Gamma_{s+}(u)\cap\Gamma_{s+}(A_n=v)} \frac{w(t)}{log|\Gamma_+(t)|}. \quad (1)$$

Here, $\Gamma_{s+}(u)$ is a set of OSN users connected to a user u and $\Gamma_{s+}(A_n = v)$ is the set of users having the attribute-value $A_n = v$. Similarly, $\Gamma_{A_n+}(u)$ is the set of all attribute-value pairs linked to user u. Therefore, the neighbourhood of u is represented as, $\Gamma_+(u) = \Gamma_{s+}(u) \cup \Gamma_{A_n+}(u)$. On the other hand, t is the set of u's friends who have an attribute-value pair $A_n = v$ (i.e. $t \in \Gamma_{s+}(u) \cap \Gamma_{s+}(A_n = v)$) and $\Gamma_+(t) = \Gamma_{s+}(t) \cup \Gamma_{A_n+}(t)$. The $w(t)$ is the weight of each of them and it's value is set to 1 in this study. The higher the value of $m(u,A_n = v)$, the higher the chance that u has the value v for attribute A_n.

For illustration, we suppose M wants to know the *emotional status* of u who considers it as sensitive and hence hides it from public view. To launch the attack, M can first prepare a data set by storing the user u's available information (for example, *Relationship status* and *Profession*) in it. Then M can visit the profiles of other users who disclose their *emotional status* as well as, additional information (i.e. *Relationship status* and *Profession* in this example) and store all these information in D. We present a sample of such data set in Table 1. The information directly related to OSN users (e.g. *Relationship status* and *Profession* as shown in Table 1) are named regular attributes, whereas the information related to the friendship links (that can be calculated by using Equation 1) are named *link attributes*. M can consider *"Emotional Status"* as the class attribute and apply any machine-learning technique to obtain the patterns of the lonely and connected users from the data set. In Figure 1, we present a sample decision forest which can be built from D (as shown in Table 1). Here, the rectangular boxes are called nodes and ovals are called leaves. The path from a root node to a leaf is called a logic rule. By using these rules, the attacker can predict the emotional status of a new user u (who is not in Table 1) even when the user does not disclose the information.

We argue that OSN users may have diverse preferences on what they consider sensitive. For example, one may consider their *emotional status* as sensitive while others may consider their *emotional status* and/or *political view* as sensitive. Therefore, a privacy-preserving technique should be capable of protecting all sensitive information for each user and should be capable of providing privacy when attackers use a different classifier than the one that the technique expects them to use. Again, a large number of attribute value suppression may provide better privacy, but at the same time, it also defeats the whole purpose of using social network sites for a user. The goal should be to provide privacy by suppressing the minimum number of attribute values.

3 OUR TECHNIQUE

The basic idea of *3LP+* is to protect the privacy of all information that a user consider sensitive. Users can give the list of attributes they considers sensitive and then, *3LP+* provides three steps (or *layers*) of recommendations:

Step 1: Compute the sensitivity of each attribute for each user and suggest to the user which attribute values the user needs to suppress.
In Step 1, as shown in Algorithm , *3LP+* selects a class attribute (from the list of sensitive attributes considered by the *3LP+* User *u*) randomly, prepares a training data set *D*, and then applies *SysFor* (Islam and Giggins, 2011) on *D* to get a set of logic rules. *3LP+* then uses the support and confidence of each rule to compute its sensitivity (or *Rule Sensitivity*) value in breaching the privacy of a sensitive attribute. Similar to previous studies (Reza et al., 2017a; Reza et al., 2017b), the rules having *Rule Sensitivity* value 1.006 or above are considered Sensitive Rules in this study. We utilize the function *GetSensitiveRules()* to represent the processes of generating the sensitive rule set R^u for *u*.

After preparing the sensitive rule set R^u, *3LP+* counts the number of appearance of each regular attribute A_n in R^u and store A_n in A^s. Here, A^s stores all the regular attributes and the number of their appearances in R^u. One attribute can appear only once in a sensitive rule R^u_j but more than once in R^u. The regular attribute A_n with the highest number of appearances in R^u is suggested to *u* for suppression. The decision is then up to *u* whether to suppress its value or not. If *u* suppresses the value of A_n, then A_n is no longer available in A^s and all sensitive rules in R^u that have A_n in their antecedent are no longer applicable for *u*. Regardless whether *u* suppresses the attribute or not, *3LP+* then identifies A_n with the next highest appearances and suggests *u* to suppress that. The process continues until R^u or A^s becomes empty.

Step 2: Hide friendship links as necessary if they are not fabricated previously.
After *Layer 1*, if any sensitive rule remains in R^u such a rule only uses link attributes (i.e. the attribute values can only be altered by using Eq. 1). Therefore, if a link attribute appears as an antecedent of a sensitive rule R^u_j (i.e. $R^u_j \in R^u$), where the value of the link attribute needs to be greater than a constant *SplitPoint* (as mentioned in R^u_j), the *3LP+* explores to reduce its value $< SplitPoint$ by hiding some of *u*'s friendship links. By doing this *3LP+* makes the rule unusable to predict the class value of *u* with certainty.

In Step 2, *3LP+* first identifies the link attribute, A_n, that appears most in the sensitive rule set R^u and compute A_n's value, denoted as V, using Eq. 1. If V is higher than the split point mentioned in R^u_j, then *3LP+* suggests *u* to hide a friendship link. While choosing a friendship link, *3LP+* selects a friend, t_i, of *u* who has the smallest degree and has not previously appeared in the friendship matrix *F* (here *F* is an $1 \times N$ matrix which stores the $Flag$ information for *u*). The *3LP+* recommends *u* to hide t_i so that it can reduce V's value the most by hiding a minimum number of friends. If *u* follows the recommendation, *3LP+* puts a $Flag$ up in the i^{th} column of the friendship matrix *F* and this ex-friend will not be recommended for further hiding or adding. *3LP+* then updates G', F', and recomputes V's value.

This process continues until the value V is lower than the *SplitPoint* mentioned in R^u_j. Once the V is lower than the split point, then the process of hiding friends stops and *3LP+* removes R^u_j and other rules (which have an antecedent with the value V) from R^u as they are no longer be applicable to determine *u*'s class value. At the end of Step 3, if R^u is not empty then only *3LP+* moves to Step 3 i.e. Layer 3.

Step 3: Add friendship links as necessary if they are not fabricated previously.
After Step 1 and Step 2, any sensitive rule remains in R^u, that contains link attribute only and tests for a value $V \leq$ *some SplitPoint* in its antecedent. In this case, *3LP+* suggests *u* to add new friends so that the V becomes greater than the split point in R^u_j and thus R^u_j is no longer applicable to *u*. While adding any friend on *u*'s friend list, a user t_i is selected in such a way that a $Flag$ has not been up previously in the i^{th} column of F' matrix and having the smallest $\Gamma_+(t)$ value. If *u* accepts the recommendation, *3LP+* then updates the matrix F', friendship graph G', and V increases. This adding process continues until the value V exceeds the split point value. It is noted that, adding a new friend on a profile is complicated and depends on the other users to confirm the friendship on OSN. Hence, *3LP+*

Algorithm 1: The Steps of 3LP+.

```
Input     : User $u$, data set $D$, friendship network $G$, total number of records $N$ in $D$, set of non-class attributes $A$, set of regular attributes $A^r$, set of link
            attributes $A^l$ where $A^r, A^l \subset A$, set of class attributes $C$.
Output    : Recommendations for $u$.
Variables : $R$ = set of sensitive rules, $R_j$ = the $j^{th}$ sensitive rule, $A_n$ = the $n^{th}$ attribute, and $F = 1 \times N$ matrix stores $Flag$ information for $u$ /*Initially all
            values in $F$ are set to $False$ */.

Step 1: Compute Sensitivity of Each Attribute for a User and Suggest the User to Suppress Attribute Values as Necessary.
    $R^u \leftarrow GetSensitiveRulesForUser(D,A,C,u)$
    foreach $R^u_j \in R^u$ do
        $n = 0$ /* The value of $n$ is always reset to 0 before the initiation of $While$ loop */
        while $n < |A|$ do
            if $A_n \in A^r$ AND $A_n$ is in the antecedent of $R^u_j$ then
                $A^s \leftarrow A^s \cup \{A_n\}$ /* Add $A_n$ in an array $A^s$ */
                $Counter_n \leftarrow Counter_n + 1$ /* Counts the number of appearance of $A_n$ in $A^s$ */
            $n = n + 1$
    while $R^u \neq \phi$ AND $A^s \neq \phi$ do
        $n \leftarrow maxarg(Counter, A)$ /* Returns the index of the attribute that appears the most */
        SuggestSuppress($A_n$) /* Suggest $u$ to suppress the value of $A_n$ */
        if $A_n$ is suppressed then
            $A^s \leftarrow A^s \setminus \{A_n\}$
            $R^u \leftarrow R^u \setminus \{R^u_j\}$ /* Rules using $A_n$ are removed from $R^u$ */
Step 2: Hide Friendship Links as Necessary if they are not fabricated previously.
    $G' = G$ and $F' = F$
    $n \leftarrow FindIndexMostSensitive(A^l, R^u)$ and $V \leftarrow CalculateValue(A_n, u)$ /* using Equation 1 */
    while $R^u \neq \phi$ AND $A^l \neq \phi$ do
        for $j = 1$ to $|R^u|$ do
            if $A_n$ is in the antecedent of $R^u_j$ AND $V \geq SplitPoint(R^u_j, A_n)$ then
                while $V \geq SplitPoint(R^u_j, A_n)$ AND $MoreFriends(u, G', F', A_n)$ do
                    $i \leftarrow FriendWithLeastDegree(u, G', F', A_n)$ /* $i$ is the index of the Friend with least degree when $F'_i \in F'$ is $False$ */
                    SuggestHide($t_i$) and $G' \leftarrow HideLink(G', u, t_i)$ /* $t_i$ is the $i^{th}$ friend */
                    $F' \leftarrow Flag(F', t_i)$ /* $F'_i$ is turned to $True$ */
                    $V \leftarrow CalculateValue(A_n, u)$
                $R^u \leftarrow R^u \setminus \{R^u_j\}$ /* Rules using $A_n$ are removed from $R^u$ */
            j=j+1
        $A^l \leftarrow A^l \setminus \{A_n\}$
        $n \leftarrow FindIndexMostSensitive(A^l, R^u)$ and $V \leftarrow CalculateValue(A_n, u)$
Step 3: Add Friendship Links as Necessary if they are not fabricated previously.
    $n \leftarrow FindIndexMostSensitive(A^l, R^u)$ and $V \leftarrow CalculateValue(A_n, u)$ using Equation 1 */
    while $R^u \neq \phi$ AND $A^l \neq \phi$ do
        for $j = 1$ to $|R^u|$ do
            if $A_n$ is in the antecedent of $R^u_j$ AND $V \leq SplitPoint(R^u_j, A_n)$ then
                while $V \leq SplitPoint(R^u_j, A_n)$ AND $MoreUsers(G', F', A_n)$ do
                    $t_i \leftarrow UserWithLeastDegree(G', F', A_n)$ /* $i$ is the index of the User with least degree when $F'_i \in F'$ is $False$ */
                    SuggestAdd($t_i$) and $G' \leftarrow AddLink(G', u, t_i)$ /* $t_i$ is the $i^{th}$ user */
                    $F' \leftarrow Flag(F', t_i)$ /* $F'_i$ is turned to $True$ */
                    $V \leftarrow CalculateValue(A_n, u)$
                $R^u \leftarrow R^u \setminus \{R^u_j\}$ /* Rules using $A_n$ are removed from $R^u$ */
            j=j+1
        $A^l \leftarrow A^l \setminus \{A_n\}$
        $n \leftarrow FindIndexMostSensitive(A^l, R^u)$ and $V \leftarrow CalculateValue(A_n, u)$
```

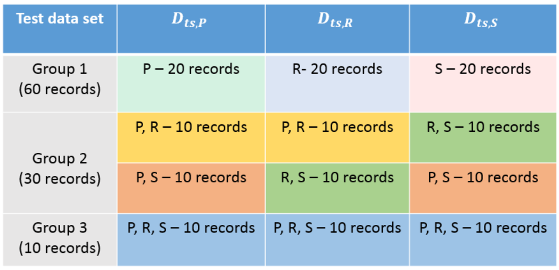
keeps these recommendations as a last resort. Our experimental results also indicate this step is seldom required.

4 EXPERIMENTS

4.1 Data Set

We implement the privacy techniques on a synthetically generated OSN data set (Nettleton, 2015) and denote it as D. The data set contains 1000 records, 11 regular attributes, and 50,397 friendship links among the users. In order to insert users' link attribute values into the data set, we calculate metric values for each regular attribute (using Equation (1)) and therefore, the total number of attributes becomes 22 (i.e. 11 regular attributes and 11 link attributes).

We prepare three versions of D for the experimental purposes for three different class attributes and they are: *political orientation* denoted as D_P,



Test data set	$D_{ts,P}$	$D_{ts,R}$	$D_{ts,S}$
Group 1 (60 records)	P – 20 records	R- 20 records	S – 20 records
Group 2 (30 records)	P, R – 10 records	P, R – 10 records	R, S – 10 records
	P, S – 10 records	R, S – 10 records	P, S – 10 records
Group 3 (10 records)	P, R, S – 10 records	P, R, S – 10 records	P, R, S – 10 records

Figure 2: Distribution of 100 test data set records in each fold.

religious view denoted as D_R, and *sexual orientation* denoted as D_S. When we consider a particular attribute as a class attribute then rest of the attributes are considered as non-class attributes. For example, in D_P data set, both *religious view* and *sexual orientation* are considered as non-class attributes.

We follow 10-fold cross validation method through out our experiments. Therefore, in each fold, a training data set, D_{tr}, contains 900 records and a testing data set, denoted as D_{ts}, contains 100 records (i.e. 10% of the total records). In order to justify the efficiency of our technique, we again split these 100 test data records, in each fold, into three different groups based on the different percentage of records. We present these three groups and their records distribution in Figure 2.

The Group 1 consists of 60 users who consider any single attribute (i.e. either "*political orientation*" or "*religious view*" or "*sexual orientation*") as sensitive. On the other hand, we assume 30 users, in Group 2, have considered any two attributes (out of the three attributes) as sensitive. Finally, Group 3 consists of 10 users who consider all the three attributes as sensitive. While preparing a test data set e.g. $D_{ts,X}$ we select the records who consider X as a sensitive information and return all other records in training data set $D_{tr,X}$. For example, we keep 50 records in $D_{ts,P}$ (as shown in $D_{ts,P}$ column) who consider *political orientation* as sensitive and return rest of the 950 records in the training data set $D_{tr,P}$. Here the different cell colours indicate different records and the same colour represents the same records.

4.2 Experimental Set-up

We now describe the experimental set-up in three phases for three different sensitive attributes. In Phase I, we first protect the privacy for *political view*, then for *religious view* in Phase II, and finally, for *sexual orientation* in Phase III. We argue that the *3LP+* can protect privacy of all the sensitive information (which are selected by its users) regardless to the sequence of selection as a class attribute. Therefore, we also conduct experiments in an opposite sequence order but for simplicity we only describe the experimental set-up here for first sequence order.

Phase I. At first step, shown in Figure 3, we prepare a training data set $D_{tr,P}$, and a testing data set $D_{ts,P}$ from the main data set D by considering users' *political view* as a class attribute. At Step 2, we apply the two privacy preserving techniques, i.e. *3LP+* and *PrivNB*, on the insecure test data sets. Here the term 'insecure' means that the *3LP+* or *PrivNB* have not been applied previously on the test data sets and hence the users' class value can be determined by an attacker easily. The test data sets are then secured by the techniques, as shown in Step 3, denoted as $D'_{ts,P}$ and $D^*_{ts,P}$ respectively. We calculate and compare the number of insecure users exists in the insecure and secure data sets. In order to provide privacy *3LP+* and *PrivNB* modifies the data sets by hiding information/friends or adding friends. Therefore, we use two different symbols $'$ and $*$ throughout the experimental set-up to denote the modified data sets by *3LP+* and *PrivNB* techniques respectively.

A privacy provider may not determine the classifier which is going to be used by an attacker and therefore, the privacy protecting technique should be able to protect privacy against any machine learning tools. In our experiments we explore and compare the performance of *3LP+* and *PrivNB* for different classifiers such as Naïve Bayes classifier (*NB*), Support Vector Machine (SVM), and Random Forest algorithm (*RF*). In order to do that we first apply these machine learning algorithms on insecure data set $D_{ts,P}$ in Step 1 and find the number of number of insecure users in the test data set. We name it as classifiers' *accuracy* which refers to the number of users whose class value is identified by the classifiers. The larger the *accuracy* value indicates lower the privacy.

We then apply the different classifiers on secure test data sets $D'_{ts,P}$ and $D^*_{ts,P}$ in Step 3. By comparing the classifiers' *accuracy* results, in Step 1 and Step 3, we then determine which technique provides better privacy on the test data sets. The results are presented in terms of number of insecure users, denoted as t_0^s and classifiers' *accuracy*, denoted as $t_0^{c_1}$. In Step 3, we also calculate data utility (in terms of number of suppressed attribute values) in $D'_{ts,P}$ and $D^*_{ts,P}$ after applying the two privacy preserving techniques and denoted as t_3^u. In Figs. 3 - 10, we use different colours of arrows to indicate the procedure of two different privacy preserving techniques.

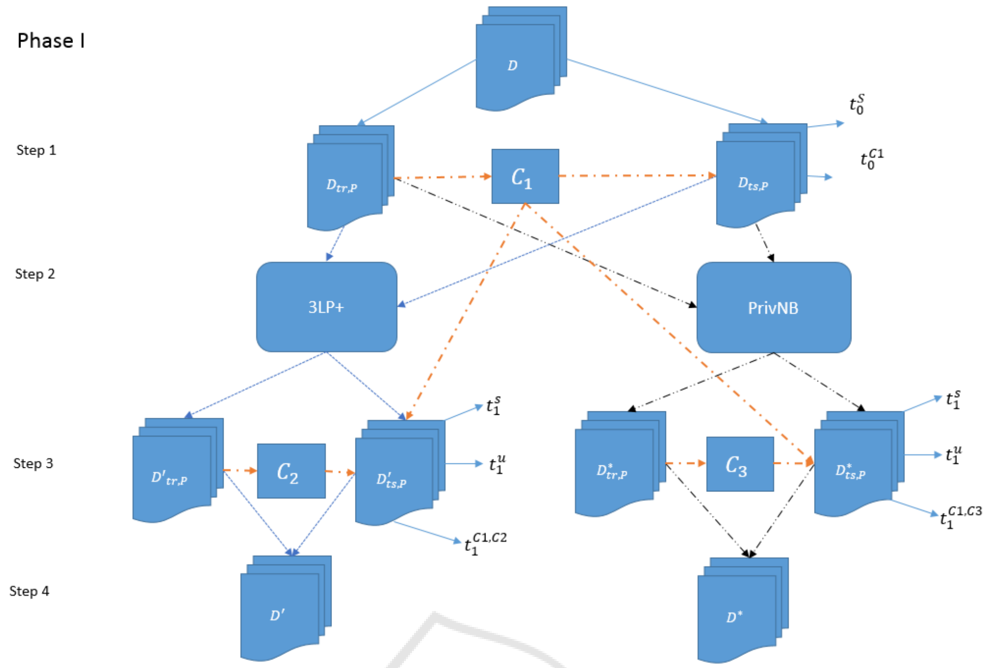


Figure 3: Phase I of the experiments.

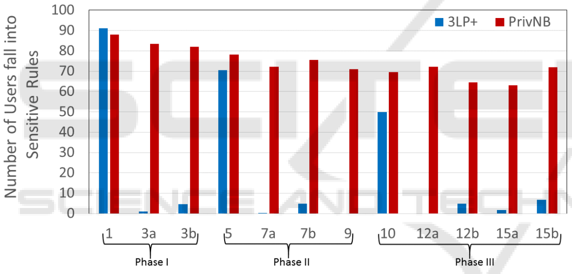


Figure 4: Prediction of class value accuracy of two privacy preserving techniques.

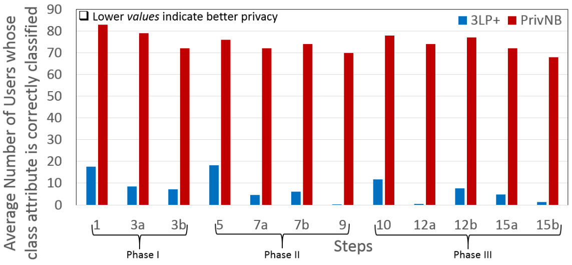


Figure 5: Prediction of users' class value(correctly) using the same classifier used by the privacy preserving techniques.

Phase II. After Phase I, we select *religious view* as a class attribute. We first prepare training and testing data sets and denote them as $D'_{tr,R}$ and $D'_{ts,R}$ which are prepared from D'. Similarly, $D^*_{tr,R}$ and $D^*_{ts,R}$ are prepared from D^*. In Step 5, different classifiers are applied on $D'_{ts,R}$ and $D^*_{ts,R}$, denoted by C_4 and C_5 respectively, to measure the classifiers' *accuracy*. Then we apply *3LP+* and *PrivNB* on test data sets in Step 7 (in order to secure the users' privacy). After Step 7 we again return all the records to original data set and thus it modified to $D^{2'}$ and D^{2*} for *3LP+* and *PrivNB* respectively. In Step 9, we again investigate the safety of users (who consider Political view as sensitive) in $D'_{ts,P}$ and $D^*_{ts,P}$ due to providing the privacy to users who consider *Religious View* again by analysing the number of insecure users. in $D^{2'}_{ts,P}$ and $D^{2*}_{ts,P}$ only.

Phase III. We select *sexual orientation* as a class attribute in this phase and similar to previous two phases, we first prepare training and testing data sets i.e. $D_{tr,S}$ and $D_{ts,S}$ as shown in Figure 10. In Step 10, we apply different classifiers, denoted by C_{10} and C_{11}, on the two test data sets $D^{2'}_{ts,R}$ and $D^{2*}_{ts,R}$ to find the classification accuracy before and applying any privacy techniques. We then apply *3LP+* on $D^{2'}_{ts,R}$ and *PrivNB* on $D^{2*}_{ts,R}$ in Step 11. We denote $D^{3'}_{ts,R}$ and $D^{3*}_{ts,R}$ to represent the secure test data sets and apply different classifiers, denoted by C_{10} and C_{11}, again on them in Step 12. After securing the test data sets, similar to Phase I and Phase II, we again analyse and compare the number of insecure users in Step 15a and Step 15b as shown in Figure 10.

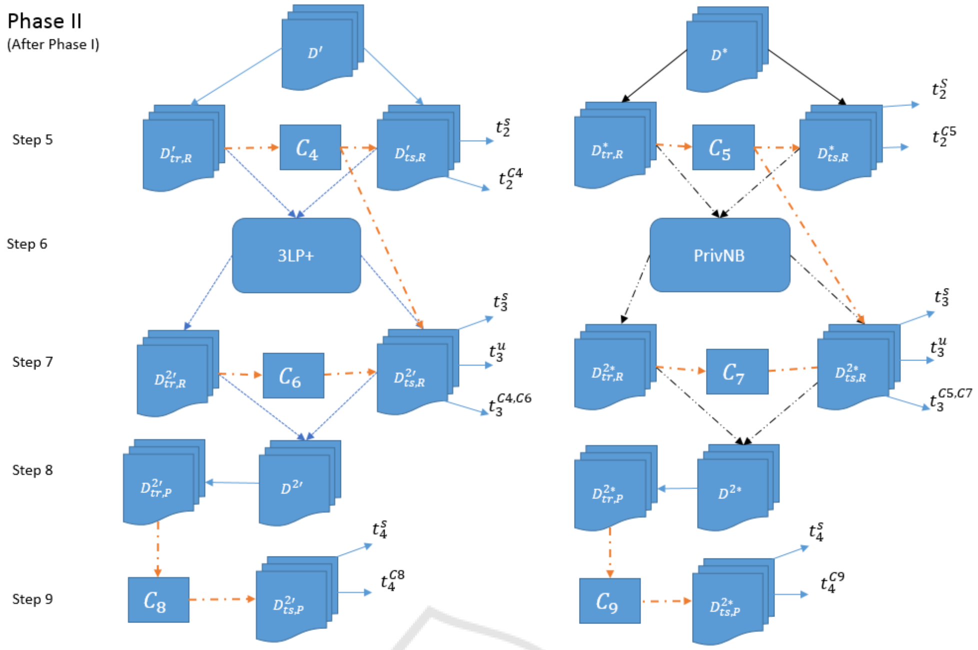


Figure 6: Phase II of the experiments.

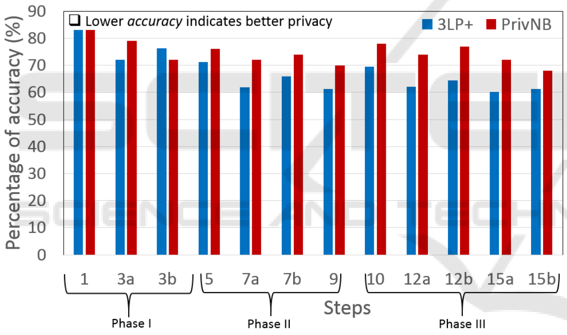


Figure 7: Performance of Naïve Bayes in order to breach users' privacy in the test data sets.

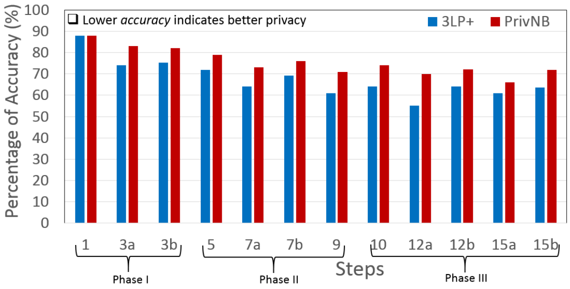


Figure 8: Performance of Support Vector Machine in order to breach users' privacy in the test data sets.

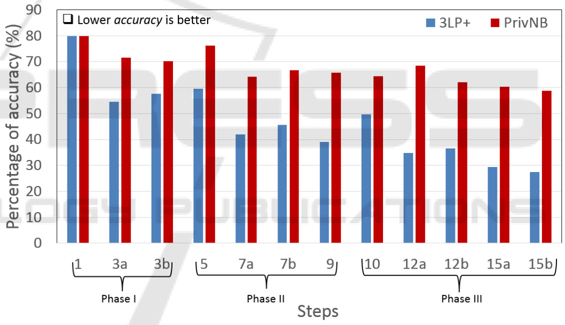


Figure 9: Performance of Random Forest Algorithm in order to breach users' privacy in the test data sets.

5 EXPERIMENTAL RESULTS

We present and compare the experimental results of two privacy preserving techniques i.e. *3LP+* and *PrivNB* in this section. The results are presented in terms of accuracy (in percentage) and the step numbers (as mentioned in the experimental set-up Section 4.2). Step number information is presented in the x-axis and accuracy information is presented in the y-axis. Here the term *accuracy* indicates the number of users whose class value is identified by the attacker. Higher accuracy indicates the greater chance for the intruder to infer the class value of a user and vice versa. In Figure 4 we present the number of insecure users whose class value can still be inferred by applying the same classifier used by the privacy protection technique. We first provide privacy by the two privacy techniques separately as described in the Section 4.2. Here y-axis represents prediction probability to classify a record (regardless to correctly or incorrectly classified) by an intruder after the protection

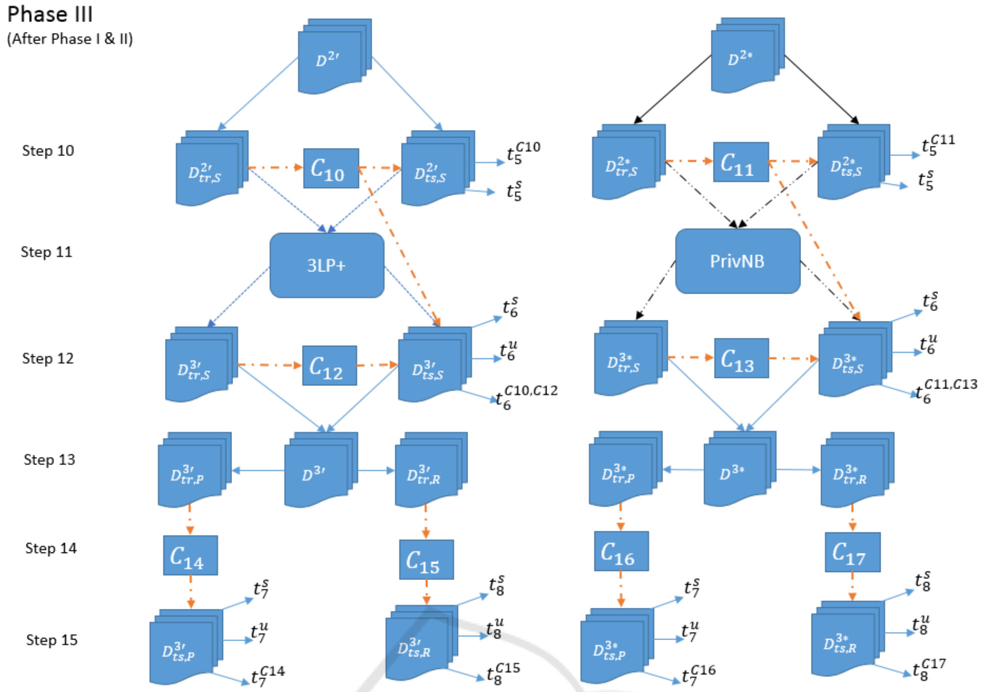


Figure 10: Phase III of the experiments.

techniques are applied. We observe that the probability percentage of records is much higher for *PrivNB* compared to *3LP+* except at Step 1. This is because the privacy preserving techniques are yet to implement at Step 1 as shown in Figure 3. On the other hand, in Figure 5, the percentage of correctly classified records by *PrivNB* is approximately 70% more than the *3LP+*.

In order to explore the performance of two privacy protection techniques against three conventional classifiers i.e. *NB*, *SVM*, and *RF*, we use the available classifier packages in WEKA. Figure 7 shows the results of using *NB* as a classification technique. We observe when we provide privacy to users in test data sets by using *3LP+*, the classification accuracy of *NB* drops about 10 percent as shown at Step 1 and Step 3. However, this accuracy drops is less than 10 percent in case of *PrivNB*. The similar trend is observed throughout the experimental steps and our technique clearly outperforms the existing *PrivNB* technique. On the other hand, a similar classification accuracy drops is observed for *SVM* and *RF* classifiers as shown in Figure 8 and Figure 9 respectively.

We also measure the data utility in terms of suppression for both techniques. In each test data set, (50 records*10 regular attribute values=500) 500 maximum attribute values are available before applying any privacy techniques. In Figure 11 we show the results for Step 3, Step 7, and Step 12 only as the privacy techniques are applied in these steps. We observe, in each step, the number of attribute value suppression by the *PrivNB* technique is three times higher than *3LP+*.

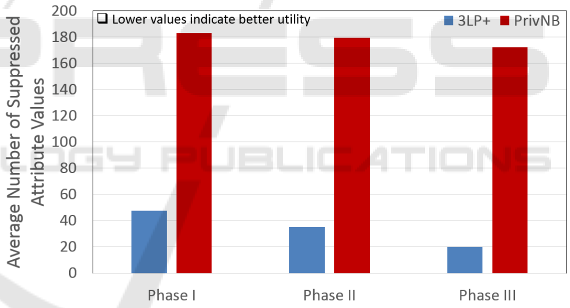


Figure 11: Comparison of attribute value suppression.

6 CONCLUSION

We propose *3LP+* in this study to provide users' privacy on social media. Previous privacy preserving techniques can protect users' single sensitive attribute (from being inferred) whereas *3LP+* can protect users' multiple sensitive attributes in one run. Our experimental results indicate that *3LP+* can provide better privacy while maintaining higher utility than an existing privacy preserving technique even if an attacker uses a different set of classifiers.

REFERENCES

Adamic, L. A. and Adar, E. (2003). Friends and neighbors on the web. *Social networks*, 25(3):211–230.

Heatherly, R., Kantarcioglu, M., and Thuraisingham, B. (2013). Preventing private information inference attacks on social networks. *IEEE Transactions on Knowledge and Data Engineering*, 25(8):1849–1862.

Islam, Z. and Giggins, H. (2011). Knowledge discovery through sysfor: a systematically developed forest of multiple decision trees. In *Proceedings of the Ninth Australasian Data Mining Conference-Volume 121*, pages 195–204. Australian Computer Society, Inc.

Johnson, C. (2009). Project gaydar. *The Boston Globe*, 20.

Nettleton, D. F. (2015). Generating synthetic online social network graph data and topologies. In *3rd Workshop on Graph-based Technologies and Applications (Graph-TA), UPC, Barcelona, Spain*.

Reza, K. J., Islam, M. Z., and Estivill-Castro, V. (2017a). 3lp: Three layers of protection for individual privacy in facebook. In *IFIP International Conference on ICT Systems Security and Privacy Protection*, pages 108–123. Springer.

Reza, K. J., Islam, M. Z., and Estivill-Castro, V. (2017b). Social media users' privacy against malicious data miners. In *Intelligent Systems and Knowledge Engineering (ISKE), 2017 12th International Conference on*, pages 1–8. IEEE.