IMPROVE - Identifying Minimal PROfile VEctors for similarity based access control

Gaurav Misra
Security Lancaster
Lancaster University, UK
Email: g.misra@lancaster.ac.uk

Jose M. Such
Security Lancaster
Lancaster University, UK
Email: j.such@lancaster.ac.uk

Hamed Balogun
Security Lancaster
Lancaster University, UK
Email: h.balogun@lancaster.ac.uk

Abstract—**There is ample evidence which shows that social media users struggle to make appropriate access control decisions while disclosing their information and smarter mechanisms are needed to assist them. Using profile information to ascertain similarity between users and provide suggestions to them during the process of making access control decisions has been put forth as a possible solution to this problem. This paper presents an empirical study aimed at identifying the minimal subset of attributes which are most suitable for being used to create profile vectors for the purpose of predicting access control decisions. We begin with an exhaustive list of 30 profile attributes and identify a subset of 2 profile attributes which are shown to be sufficient in obtaining similarity between profiles and predicting access control decisions with the same accuracy as previous models. We demonstrate that using this pair of attributes will help mitigate the challenges encountered by similarity based access control mechanisms.**

1. Introduction

Defining appropriate access controls during information disclosure is a challenging task for all social media users. The plethora of contacts most users have and the different life facets (such as work, family, etc.) represented by these contacts only add to the user's complications. Moreover, social media applications have often been found to be falling short in supporting and assisting users in making access control decisions [1]. Without adequate assistance, users often make erroneous choices when making such decisions [2]. Mainstream social media sites such as Facebook and Google+ have made an effort to mitigate this problem by assisting users in managing their friend networks by creating Lists [1] and Circles [3] respectively. However, recent research findings suggest that hardly any users employ these features when making access control decisions [4].

There are some proposed approaches which suggest that information about the content being shared (text, photos, etc.) can be used to assist users in making appropriate access control decisions [5]. This "metadata" can either be automatically generated based on analysis of the content or provided by the user in the form of annotations. Both methods suffer from the significant challenge of accurately defining the content which can then be used to assist the users. Moreover, the dynamic nature of content creation in social media necessitates real-time analysis to be quick which further accentuates the demands on mechanisms relying on automatic definition of content. Conversely, if the responsibility to annotate the content is on the user, it creates an additional burden on them.

Another approach of providing assistance to users in making access control decisions relies on creating suggestions based on learning the user's access control behavior. These suggestions rely on associations of a user's friends based on prior partitioning of the friend network [6] or similarity between their profiles [7], [8]. Such "Similarity based Access Control" rely on "Profile Vectors" which encapsulate the information stored in social media profiles of individuals in a manner which captures the similarity between them. This similarity can then be used to either group users together in communities [8], [9] or make suggestions to assist the user in selecting an audience for their content in real-time [7]. All profile similarity based access control mechanisms, however, have to overcome several limitations: **1)** The information required to create profile vectors should be processed in *minimal time* to assist the user; **2)** The information that is collected should be *easy to code into profile vectors*; **3)** There is a possibility of encountering *missing information* for some attributes as users often refrain from populating many fields on their social media profiles [7] and **4)** The information collection for the purpose of creating these profile vectors should be as *less intrusive as possible*. Thus, processing of personal information and the users' communication with their friends should be minimal in any such mechanism.

Keeping these limitations in mind, it seems imperative to identify the minimal subset of profile features which can be used to accurately predict access control decisions and make appropriate suggestions to the user. The endeavor is to create an access control mechanism which provides accurate suggestions with minimal profile information from the users or their friends. All of the previous works which have used profile information have based the choice of the set of attributes solely on previous literature or qualitative analysis

[1] https://en-gb.facebook.com/help/135312293276793/

of user responses. There is an absence of an evaluation of the available profile attributes in social network infrastructures to identify the optimal subset for the purpose of employing user similarity for access control decisions. In this paper, we bridge this gap by providing a *systematic analysis of all profile attributes available in social media profiles to select the minimal subset most suitable for predicting access control decisions with maximum possible accuracy.*

Our findings help us in identifying a subset of 2 profile attributes, namely *Total Friends* and *Mutual Friends*. We demonstrate that a classifier created using these two attributes can predict access control decisions with the *same accuracy as previous models* and improves upon them to mitigate the discussed challenges due to the following advantages:

- The time taken to simply fetch profile information relevant to these attributes using Facebook API calls is approximately *one-sixth* of the time taken to fetch information relevant to subsets of attributes from the quickest among previous works.
- Both attributes facilitate *easier creation of profile vectors* because they do not require any coding and can be directly fetched from the profiles.
- The attributes *do not rely on information such as location, workplace and age* which can often be left blank or misreported by the users.
- The attributes do not require information about the users' communication (posts, messages, etc.) and hence makes the information gathering *less intrusive as compared to previous approaches.*

2. Background and Related Work

The lack of usability of access control mechanisms in social media has attracted a lot of research in recent times. One of the methods in which users can be assisted in making appropriate access control decisions is by enhancing automatic contact grouping mechanisms. This has been acknowledged by mainstream social media sites like Facebook and Google+ who have created "Smart Lists" and Circles [3] for the users to assist them in managing and organizing their contacts. However, this has not solved the problem of defining access controls as the majority of users do not employ these mechanisms when making access control decisions [4]. This suggests that smarter solutions are required to alleviate the burden on the users.

Social media profiles are often created to contain a plethora of information about individuals. There are numerous ways in which this information can be leveraged to enhance access control mechanisms. One particular way is to use profile information to calculate relationship closeness or "tie-strength" to define relationships between individuals in a social network. Tie-strength between individuals can be measured and modeled using various factors such as network structure (community membership), extent of communication and similarity of profile data [11]. This can be used to assist users while making access control decisions. For example, a user may choose to disclose something to only his "close friends" (based on the tie-strength calculation) and deny access to "distant" or "weak" relationships. Fogues et al. [10] created BFF (Best Friends Forever) which contained a "tie-strength module" which calculates the tie-strength between individuals based on a subset of profile attributes outlined by Gilbert and Karahalios [11]. BFF chose to ignore profile attributes such as location or work as they asserted that these are often left blank by social media users and showed that the results are not substantially affected by removing these profile attributes. Our work considers these attributes as we wanted to examine as comprehensive a list of attributes as possible. We handle missing information in the way we code variables to capture the similarity between individuals as explained later in the paper.

Profile information can also be used to create profile vectors to define individuals on a social network and capture the similarity between them. This similarity can then be used to mine appropriate privacy policies for a user under the assumption that similar users will have similar privacy policies [9]. Another important usage of profile similarity is in creating "profile based grouping" mechanisms to assist users in managing their friend networks. Profile information can be coded to capture similarity between users which can be used to partition a user's friend network by grouping similar friends together [7], [8]. This method can be used to create static groups a-priori which can then be leveraged by users to make access control decisions [8]. An alternative method is to use profile similarity to provide "on-demand group creation" as shown in ReGroup [7] which provides suggestions to the user at the time of making access control decisions by learning from their previous choices. These suggestions are based on calculating the similarity between the profiles of the user's friends. However, the evaluation shown in their paper is based on qualitative feedback and measures factors such as time taken to form groups and the size of the groups created by the participants to ascertain usability. It is claimed that manually labeled dataset for examining the accuracy of the groups was unavailable for their evaluation [7]. Additionally, all previous works fail to identify the minimal set of profile attributes most suitable to predict access control decisions. We bridge these gaps with our work which uses empirical data to identify the most suited set of profile features which can be used to predict access control decisions for individuals. We use a list of 30 profile attributes, constructed from all the available profile information using the Facebook API, as a basis for our analysis. We show in Table 1 that we consider all the attributes considered by some of the above mentioned approaches (we could not compare our list with works which did not explicitly provide the list of profile attributes they used) as well as some extra attributes available in social media infrastructure in addition to them.

TABLE 1. Comparison of the subsets of profile attributes used to create profile vectors in the different approaches

No.	**Profile Feature**	**Type**	**Our Work**	**Social Circles** [8]	**ReGroup** [7]	**BFF** [10]
1	Friendship Duration	Derived	✔	✔	✔	✔
2	Recency of Communication	Derived	✔	✔	✔	✔
3	Amount seen together (photos together)	Derived	✔	✔	✔	✔
4	Wall Messages	Derived	✔	✔	✔	✔
5	Inbox Messages	Derived	✔	✔	✔	✔
6	Mutual Friends	Direct	✔	✔	✔	✔
7	Gender	Direct	✔	✔	✔	✖
8	Age	Direct	✔	✔	✔	✖
9	Family Membership	Direct	✔	✔	✔	✖
10	Home Town	Direct	✔	✔	✔	✖
11	Home State	Direct	✔	✔	✔	✖
12	Home Country	Direct	✔	✔	✔	✖
13	Current City	Direct	✔	✔	✔	✖
14	Current State	Direct	✔	✔	✔	✖
15	Current Country	Direct	✔	✔	✔	✖
16	Education	Direct	✔	✔	✔	✔
17	Work	Direct	✔	✔	✔	✖
18	Likes	Direct	✔	✔	✖	✔
19	Events	Direct	✔	✔	✖	✖
20	Politics	Direct	✔	✔	✖	✖
21	Religion	Direct	✔	✔	✖	✖
22	Interests	Direct	✔	✔	✖	✖
23	Links Shared	Derived	✔	✖	✖	✔
24	Music	Direct	✔	✖	✖	✖
25	Movies	Direct	✔	✖	✖	✖
26	Languages	Direct	✔	✖	✖	✖
27	Sports	Direct	✔	✖	✖	✖
28	Total Friends	Direct	✔	✖	✖	✔
29	Friend Difference	Direct	✔	✖	✖	✖
30	Age Difference	Direct	✔	✔	✔	✖

3. Method

This research experiment was conducted at Lancaster University. Participants were invited via email, posters and word of mouth. Details about the privacy implications and the overall objectives of the project were communicated to the registered participants as per the guidelines prescribed by the Ethics department at the Research Support Office (RSO) in Lancaster University[2]. After appropriate ethics clearance, participants were invited to use a Facebook application designed and developed by the researchers specifically for this research project.

3.1. Experiment

The application used for the experiment was built using the Facebook API to fetch information from the participants' profile and their friend connections. All this data was then stored in secure databases for subsequent analysis.

Five photos were randomly downloaded from Facebook profiles of the participants by the application to be presented to the user for making access control decisions. In addition, the participants were asked to select and bring five other photos which they hadn't previously uploaded on Facebook. This was done to avoid a scenario where a user selects an audience for a photo during the study for which they had already received comments and likes as that may have influenced their choice of audience members. The participants were also advised to choose photos which were personal (either included them or a family member) or considered sensitive so that they had a privacy implication. Thus, in total, each participant had to make access control decisions for 10 photos. The different stages of the user study were:

1) The participants logged into the application using their Facebook credentials. They were then alerted about the data the application would access and asked for explicit consent before moving on.

2) They were shown 10 photos (5 from Facebook and 5 they brought as detailed earlier) sequentially on the screen, each on an individual page. The participants were then asked to select an audience for the photo from an alphabetically sorted friend list to imitate the organization Facebook uses to show friend lists to its users. The friends were not grouped in any way to avoid bias (even preset groups such as Facebook 'lists' were ignored). They were mandated to choose at least one audience member for each photo before progressing to the next step. The participants were instructed to select each and every friend that they would want to grant access to the photo. They were also explicitly told that any friend who is not selected

[2] http://www.lancaster.ac.uk/fass/resources/ethics/procedures.htm

would be denied access to the photo.

Once the participants selected all the audiences, these access control decisions were stored in a secure database to be used later as the ground truth for the analyses. The detailed list of profile attributes that were obtained by coding the information collected during the study are shown in Table 1.

Due to the Facebook API[3], only the profile information of users who use an application (and explicitly provide data access permissions) is available for collection. To mitigate this, we particularly encouraged groups of people to participate in the study so that a particular participant would have some Facebook friends also participating in the study which would enable us to get their profile information. The users were not informed as to which of their friends' information we would be able to access. The developed application asked the users' consent to access information of ALL their Facebook friends as we did not want the users to pay special attention towards their friends who were fellow participants while making their access control decisions. We particularly had 23 participants in total out of which 14 (61%) were female. The average age of the participants was 32 years (s.d=10, max=61, min=23). Considering each available friend profile of the 23 participants (2-5 friends per participant) and all photos for which they selected audiences, our dataset contained a total of *689 access control decisions*. This constituted our final manually labeled dataset which is used as ground truth for the analysis of the profile attributes.

3.2. Coding the Profile Features

The profile information collected during the user study had to be converted into profile vectors for each user and their friends to capture the similarity between them. We aimed to get as many profile attributes as possible using the Facebook API [4]. We considered all the fields which consist of information about the individual but chose to ignore descriptive fields such as "About Me" and "User Bio". We also did not include "Relationship Status" and "Significant Other" as none of our participants had disclosed them on their profiles. The attributes listed in Table 1 are categorized in the following two types:

- *Direct*: These attributes can be directly fetched from the Facebook profile of the user or the friend. No calculation or aggregation is required.

- *Derived*: These attributes are created by aggregating different forms of communication (for eg: posts, messages or photos) between a user and a particular friend.

22 out of the 30 profile attributes shown in Table 1 are *Direct* attributes and are fetched directly from the Facebook profiles. Intuitively, the computational cost of creating profile vectors is minimized if a large number of attributes are *Direct* as opposed to *Derived*. Thus, the method of obtaining the attribute is an important factor to consider while evaluating the profile attributes.

The coding of profile features in our work is most similar to [8] as they also looked to create vectors based on similarity of profiles though we consider some additional attributes which were not included by them as shown in Table 1. The various attributes were coded in different ways depending on the type of information they contain in order to capture the similarity between a user and his friends.

- *Friendship Duration* and *Recency of Communication* between a particular user and a specific friend were calculated by counting the number of days since the first and latest communication respectively.

- *Amount seen together, Wall messages* and *Inbox messages* are all coded by simply counting the number of interactions shared by the user and a particular friend. A zero value indicates that no interaction took place between them during the time for which the data was collected.

- *Mutual friends, Links shared* and *Events* denote the common friends, links and events attended by a user and a particular friend.

- *Gender* was coded as a binary variable. Here, '1' indicates that the user and the friend had the same gender while a '0' indicates a dissimilarity.

- *Age* and *Total friends* were taken directly from the users' profiles. We also calculated the difference (absolute value) between the user's age and a friend's age as a new variable called *Age Difference*. This gives us a measure of the gap between the user and a potential audience member in terms of their age. A similar calculation was done for Total Friends to create the variable *Friend Difference*.

- *Family membership* was coded as a binary variable where a '1' indicates that the particular friend is disclosed as a family member of the user while a '0' indicates that no such family relationships have been found for the friend and the user. This includes cases where the user did not disclose family relationships on Facebook.

- The attributes numbered *10 to 27* in Table 1 were coded to represent the number of common entries. For example, if a user and a friend had exactly one common educational institution, a '1' value was put for that variable. A '0' value captures both non-matches and missing entries. Missing entries were coded as 0 to ensure that we maximize the effect of matches between the values while capturing the similarity between profiles.

[3] https://developers.facebook.com/docs/apps/upgrading

[4] https://developers.facebook.com/docs/graph-api/reference/user

TABLE 2. CLASSIFIER CONSIDERING ALL 30 PROFILE ATTRIBUTES SHOWN FOR EACH ALGORITHM

Condition	Classification Algorithm	Class	Instances	TP Rate	FP Rate	Precision	Recall	F-Measure
Unfiltered Data	**Naive Bayes**	**1**	211	0.597	0.219	0.450	0.597	0.514
		0	478	0.781	0.403	0.866	0.781	0.821
		Total	689	0.739	0.360	0.770	0.739	0.750
	Support Vector Machines	**1**	64	0.365	0.011	0.906	0.365	0.520
		0	625	0.989	0.635	0.838	0.989	0.907
		Total	689	0.845	0.491	0.854	0.845	0.818
	Random Forest	**1**	135	0.642	0.062	0.756	0.642	0.694
		0	554	0.938	0.358	0.897	0.938	0.917
		Total	689	0.869	0.290	0.864	0.869	0.865
Spread Subsample Filter	**Naive Bayes**	**1**	206	0.585	0.213	0.451	0.585	0.510
		0	483	0.787	0.415	0.863	0.787	0.823
		Total	689	0.740	0.369	0.768	0.740	0.751
	Support Vector Machines	**1**	70	0.371	0.021	0.843	0.371	0.515
		0	619	0.979	0.629	0.838	0.979	0.903
		Total	689	0.839	0.489	0.839	0.839	0.814
	Random Forest	**1**	132	0.604	0.068	0.727	0.604	0.660
		0	557	0.932	0.396	0.887	0.932	0.909
		Total	689	0.856	0.320	0.850	0.856	0.851
Class Balancer Filter	**Naive Bayes**	**1**	325.4	0.654	0.291	0.692	0.654	0.673
		0	363.6	0.709	0.346	0.672	0.709	0.690
		Total	689	0.682	0.318	0.682	0.682	0.682
	Support Vector Machines	**1**	312.4	0.692	0.215	0.763	0.692	0.726
		0	376.6	0.785	0.308	0.718	0.785	0.750
		Total	689	0.738	0.262	0.740	0.738	0.738
	Random Forest	**1**	341	0.748	0.240	0.757	0.748	0.753
		0	348	0.760	0.252	0.751	0.760	0.756
		Total	689	0.754	0.246	0.754	0.754	0.754
Cost Sensitive Classifier (Cost = 2)	**Naive Bayes**	**1**	239	0.616	0.266	0.410	0.616	0.492
		0	450	0.734	0.384	0.864	0.734	0.794
		Total	689	0.707	0.357	0.760	0.707	0.724
	Support Vector Machines	**1**	111	0.509	0.057	0.730	0.509	0.600
		0	578	0.943	0.491	0.865	0.943	0.903
		Total	689	0.843	0.390	0.834	0.840	0.833
	Random Forest	**1**	168	0.673	0.115	0.637	0.673	0.654
		0	521	0.885	0.327	0.900	0.885	0.892
		Total	689	0.836	0.278	0.839	0.836	0.838

4. Results

The data collected during the user study included the profile information of the participants and their friends as well as the access control decisions taken by the participants during the experiment. The profile information was coded, as has been described, to create profile attributes to be used to ascertain similarity between profiles. The access control decisions made by the participants were coded as ‘1’ (for “allow”) and ‘0’ (for “deny”). Recall that we want to identify a subset of profile attributes which are most suitable for predicting access control decisions. We start by using the list of all 30 profile attributes to create the classifier before moving on to identify the minimal subset of profile attributes most suitable for predicting access control decisions.

4.1. Prediction of Access Control Decisions

We tried 3 different classification algorithms: *Naive-Bayes* [12], *Support Vector Machines* [13] and *Random Forest* [14]. All algorithms were implemented using Weka [15] with 10 fold cross validation. We started with a baseline condition of using all the 30 profile attributes in the classifier.

Our dataset consisted of *689* access control decisions from 23 users. Out of this, *159* were *“allow” (coded as class ‘1’)* while *530* were *“deny” (coded as class ‘0’)*. Thus, we observed some imbalance between the two classes and decided to implement different well-established class balancing methods [16] to understand if they could have an effect on the accuracy of the classifier. The different modes of operation of the classifier shown in Table 2 are:

- *Unfiltered*: The complete input data consisting of all 689 decisions was considered “as-is” without any filtering in this setting.
- *Spread Subsample Filter*: This is a filter implemented in Weka which allows maximum “spread” between the two classes in our analysis. This method does not create artificial instances of the rarer class (‘1’ in our case) but simply redistributes the frequency in a way which balances the classes.
- *Class Balancer Filter*: This method introduces synthetic instances of the rarer class to balance the dataset. This is also denoted by the fractional numbers in the “Instances” column of Table 2.

- *Cost-Sensitive Classifier*: We employed "Cost sensitive learning" [17], [18] which penalizes any instance of class '1' classified as class '0' in our case. After several iterations, we found that selecting a cost value of 2 provided the best results. Increasing cost further would result in worse overall accuracy as it increases the false positive rate.

The results in Table 2 show that Random Forest was the best performing algorithm as it has the highest overall F-measure for all conditions. We see that using the Class Balancer filter improves accuracy of the '1' class but reduces the overall accuracy of the Random Forest classifier. However, it is interesting to note that for Naive-Bayes classifier, the *Spread Subsample* filter has a marginally higher accuracy (0.751) as compared to the unfiltered setting (0.750). Similarly, for SVM we observe that using the *Cost Sensitive Classifier* (0.833) produces higher accuracy than the default setting (0.818). Overall, we can observe that Random Forest is the most suitable classification algorithm for our data and we use it to perform further analysis in the rest of the paper. The best accuracy provided by Random Forest is in the *unfiltered* setting which is *86.5%*. We use this as a baseline accuracy provided by a classifier using all 30 attributes.

4.2. IMPROVE

The main objective of the research was to identify the minimal profile vector to be used for similarity based access control. We used Weka [15] to calculate the "information gain" for each profile attribute to understand how they contribute to the classifier. The 30 profile attributes are ranked according to the information gain value in Table 3. We can see from the table that *Amount Seen Together* has the highest information gain (0.213). There are 10 profile attributes which do not provide any information gain (attributes 21-30 in Table 3).

In order to identify the minimal subset of profile attributes, we selected the first 5 profile attributes with the highest information gain. These are *Amount Seen Together (0.213), Total Friends (0.190), Mutual Friends (0.182), Friendship Duration (0.175)* and *Likes (0.174)* (see Table 3). We systematically created classifiers with all possible combinations of these 5 profile attributes using Random Forest algorithm with the aim of producing an accuracy of *86.5%* that would match that of the classifier created with all 30 attributes as shown earlier. We started by using all 5 identified attributes to create a classifier and found that it produced the desired accuracy of 86.5%. We then proceeded to try all possible permutations and combinations of these 5 attributes to try and find a minimal set which matches this maximum accuracy (86.5%). The only combinations of less than 5 attributes that produced the same accuracy of 86.5% were *(Total Friends,Mutual Friends)* and *(Mutual Friends, Likes)*. We did not find any attribute which could produce the same accuracy when taken individually.

TABLE 3. PROFILE ATTRIBUTES RANKED ACCORDING TO INFORMATION GAIN

No.	Profile Attribute	Information Gain
1	Amount seen together	0.213
2	Total Friends	0.190
3	Mutual Friends	0.182
4	Friendship Duration	0.175
5	Likes	0.174
6	Friend Difference	0.154
7	Age Difference	0.107
8	Movies	0.102
9	Links Shared	0.096
10	Wall Messages	0.071
11	Events	0.067
12	Interests	0.050
13	Recency of Communication	0.045
14	Home Town	0.037
15	Age	0.036
16	Current State	0.034
17	Current City	0.032
18	Home State	0.026
19	Gender	0.025
20	Family	0.008
21	Work	-
22	Home Country	-
23	Current Country	-
24	Music	-
25	Languages	-
26	Sports	-
27	Religion	-
28	Politics	-
29	Inbox Messages	-
30	Education	-

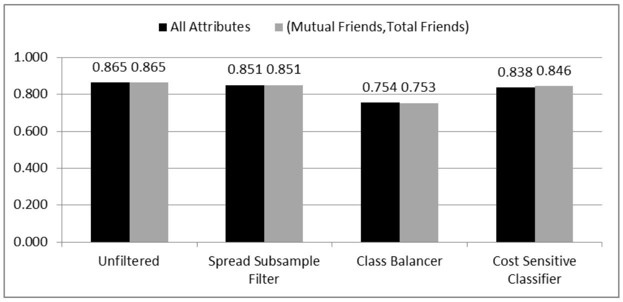

Figure 1. Comparison of overall F-measures for classifier with all attributes and classifier with the pair of identified attributes for all modes of operation using Random Forest algorithm

When we compare these two pairs of profile attributes, we find that they both contain *Mutual Friends*. Thus, we have a choice between *Total Friends* and *Likes* in order to finalize the selection of attributes. Considering the process of coding these two variables to create profile vectors, we find that *Total Friends* can be directly fetched as a numeric value from a user's profile and used "as-is" in the profile vector. Alternatively, in order to code *Likes* to capture similarity between two users, the list of pages liked by them needs to retrieved and compared to count the number of pages liked by both of them. This resultant value is then used in the profile vector. Thus, considering the comparatively more complex coding procedure for *Likes* as compared to *Total Friends*, we select **(Mutual Friends,Total Friends)**

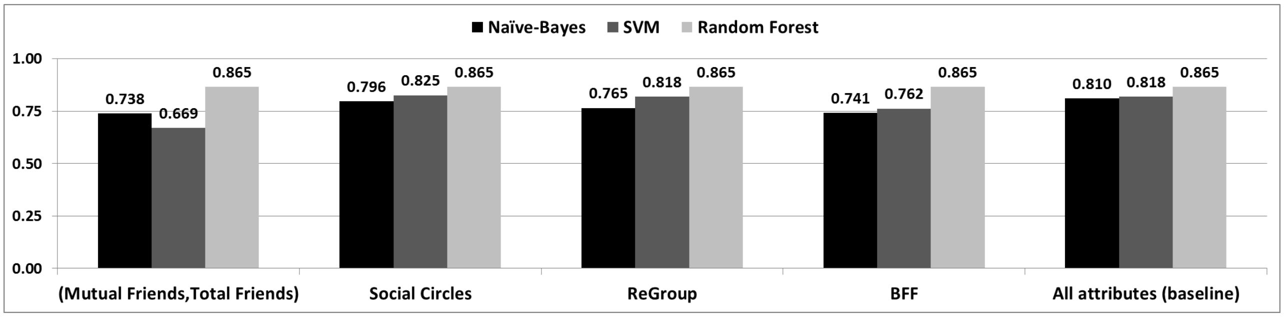

Figure 2. Comparison of overall F-measures for all classifiers (unfiltered data) listed per algorithm

as the *minimal set of profile attributes which can be used to accurately predict access control decisions*.

We also tested the classifier created by using the pair of attributes *(Mutual Friends,Total Friends)* in all the operating modes to account for the class imbalance as we had done with the entire list of 30 attributes. The comparison between the performance of the classifier created using these two attributes and that created with all 30 attributes is shown in Figure 1. It can be seen from the figure that the unfiltered dataset performs best with Random Forest algorithm for the entire set of 30 attributes as well as the identified subset of two profile attributes.

4.3. Comparison with Previous Approaches

We show the comparison of overall F-measure values for our classifiers (both baseline and enhanced setting using only 2 attributes) with the profile attributes used by *Social Circles* [8], *ReGroup* [7] & *BFF* [10] in Figure 2. It can be seen that all classifiers produce identical performance for Random Forest algorithm and this also has the highest accuracy (86.5%). Thus, we can clearly see that our identified pair of profile attributes is sufficient in replicating the accuracy of previous models for the best performing algorithm and adding further attributes to these 2 does not improve the accuracy of the classifier at all.

Another important factor to be considered while evaluating profile attributes in our dataset is the temporal cost on a system that would employ these profile vectors to try and predict access control decisions. This would include the time taken to fetch the profile information from Facebook and then the time and computation required to code this information to create profile vectors as described earlier in section 3. We show a comparison of the estimated time[5] taken to fetch the profile information necessary to create the profile vectors containing *(Mutual Friends,Total Friends)* and the subset of profile attributes used by *Social Circles* [8], *ReGroup* [7] & *BFF* [10] in Figure 3. The time was calculated by creating 50 API calls to fetch the corresponding subset of profile attributes and taking the average for each proposed mechanism. As can be seen from the figure, using our identified pair *(Mutual Friends,Total Friends)* takes far less time (about *one-sixth* of the next best) than it would take to fetch attributes corresponding to the other approaches. This is mainly due to the fact that our approach does not use any communication information (such as wall messages, inbox messages, etc.) which require more time as compared with "direct" profile attributes such as mutual friends and total friends, which can be easily fetched from the profiles themselves. Moreover, additional time will be required to process the fetched information in the case of most of the "derived" attributes which would further magnify the difference in the temporal cost between the approaches.

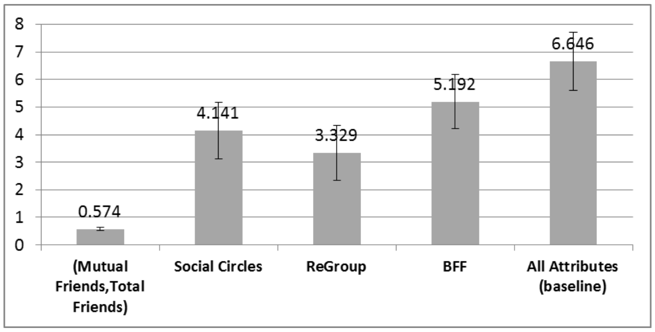

Figure 3. Time (in seconds) required to fetch profile information using Facebook API corresponding to each subset of profile attributes

5. Discussion

We iterated through different configurations of the classifier by using various combinations of the 5 top ranked profile attributes with the highest information gain. We found that using *Mutual Friends* and *Total Friends* was sufficient to replicate the maximum accuracy (86.5%) of the classifier created with all 30 attributes and that of previous related works. We have also demonstrated that this subset of attributes enhances previous profile similarity based access control models in several ways:

Less time to fetch required information. We have shown that the time required to fetch all relevant information from Facebook profiles for the 2 attributes identified in this study is about *one-sixth* of the time required for fetching

[5] We ran API calls for a Facebook profile with 320 friends with default settings

information required for the subset of profile attributes corresponding to the quickest among previous works.

Easy creation of profile vectors. The "direct" attributes *Total Friends* and *Mutual Friends* are both simply numeric values which can be directly fetched from the users' profiles and incorporated in the profile vector without the need of processing a lot of information. There are certain attributes like *Education, Workplace,etc.* who can have different names for the same entity which makes coding more difficult. For both *Total Friends* and *Mutual Friends*, coding is unambiguous as these are simply numeric values.

Less intrusive information gathering. The attributes do not rely on communication (messages, wall posts, etc.) between individuals and hence can be considered less intrusive than other models which use such information. We showed earlier that all previous models evaluated by us use this information to create the profile vectors.

Cannot be left blank or faked. Both *Total Friends* and *Mutual Friends* are attributes that are automatically updated by the social media site itself. The users cannot manipulate this data in any way. These attributes do not require access to any identifying information such as *Gender, Address, Age, Workplace, etc.,* which are often left blank by users on their profiles.

6. Conclusion

This paper presents an analysis of profile attributes using empirical data and identifies a pair of profile attributes which is found to be sufficient to calculate user similarity and predict access control decisions. We show that a classifier created with these attributes can replicate the accuracy of classifiers based on sets of attributes used by previous works in this area. Using these profile attributes to predict access control decisions has several advantages and can contribute towards mitigating some of the limitations of profile similarity based access control mechanisms. The identification of these attributes can contribute effectively towards the design and implementation of future access control mechanisms.

An interesting future work is to investigate the effect of the nature of the content (photos, posts, etc.) and evaluate whether this information can be used to augment the profile vector based approach for further accuracy as well as the use of the 2 attributes identified in this study in an interactive setting like ReGroup [7].

References

[1] G. Misra and J. M. Such, "How socially aware are social media privacy controls?" *Computer*, vol. 49, no. 3, pp. 96–99, 2016.

[2] G. Hull, H. R. Lipford, and C. Latulipe, "Contextual gaps: Privacy issues on facebook," *Ethics and information technology*, 2011.

[3] S. Kairam, M. Brzozowski, D. Huffaker, and E. Chi, "Talking in circles: selective sharing in google+," in *Proc. of the SIGCHI*. ACM, 2012.

[4] P. Wisniewski, B. P. Knijnenburg, and H. Richter Lipford, "Profiling facebook users privacy behaviors," in *SOUPS2014 Workshop on Privacy Personas and Segmentation*, 2014.

[5] A. C. Squicciarini, S. Sundareswaran, D. Lin, and J. Wede, "A3p: adaptive policy prediction for shared images over popular content sharing sites," in *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*. ACM, 2011, pp. 261–270.

[6] L. Fang and K. LeFevre, "Privacy wizards for social networking sites," in *Proc. of the Conference on World wide web*. ACM, 2010.

[7] S. Amershi, J. Fogarty, and D. Weld, "Regroup: Interactive machine learning for on-demand group creation in social networks," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2012, pp. 21–30.

[8] J. J. McAuley and J. Leskovec, "Learning to discover social circles in ego networks." in *NIPS*, vol. 2012, 2012, pp. 548–56.

[9] A. Squicciarini, S. Karumanchi, D. Lin, and N. DeSisto, "Identifying hidden social circles for advanced privacy configuration," *Computers & Security*, vol. 41, pp. 40–51, 2014.

[10] R. L. Fogués, J. M. Such, A. Espinosa, and A. Garcia-Fornes, "Bff: A tool for eliciting tie strength and user communities in social networking services," *Information Systems Frontiers*, 2014.

[11] E. Gilbert and K. Karahalios, "Predicting tie strength with social media," in *Proc. of the SIGCHI*. ACM, 2009.

[12] K. P. Murphy, "Naive bayes classifiers," *University of British Columbia*, 2006.

[13] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf, "Support vector machines," *Intelligent Systems and their Applications, IEEE*, vol. 13, no. 4, pp. 18–28, 1998.

[14] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[15] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[16] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent data analysis*, 2002.

[17] C. X. Ling and V. S. Sheng, "Cost-sensitive learning," in *Encyclopedia of Machine Learning*. Springer, 2011, pp. 231–235.

[18] C. Elkan, "The foundations of cost-sensitive learning," in *International joint conference on artificial intelligence*, vol. 17, no. 1. Lawrence Erlbaum Associates Ltd, 2001, pp. 973–978.