Soc. Netw. Anal. Min.
DOI 10.1007/s13278-012-0051-2

REVIEW ARTICLE

Enabling cross-site interactions in social networks

Mohamed Shehab · Moonam Ko · Hakim Touati

Received: 19 April 2011 / Revised: 17 November 2011 / Accepted: 12 January 2012

Abstract Online social networks is one of the major technological phenomena on the Web 2.0. Hundreds of millions of people are posting articles, photos, and videos on their profiles and interacting with other people, but the sharing and interaction are limited within a same social network site. Although users can share some contents in a social network site with people outside of the social network site using a secret address of content, appropriate access control mechanisms are still not supported. To overcome this limitation, we propose a cross-site interaction framework *x-mngr*, allowing users to interact with users in other social network sites, with a cross-site access control policy, which enables users to specify policies that allow/deny access to their shared contents across social network sites. We also propose a partial mapping approach based on a supervised learning mechanism to map user's identities across social network sites. We implemented our proposed framework through a photo album sharing application that shares user's photos between Facebook and MySpace based on the cross-site access control policy that is defined by the content owner. Furthermore, we provide mechanisms to enable users to fuse user-mapping decisions that are provided by their friends or others in the social network. We implemented our framework and through extensive experimentation we prove the accuracy and precision of our proposed mechanisms.

Keywords Social networks · Profile mapping · Supervised learning

M. Shehab (✉) · M. Ko · H. Touati
Department of Software and Information Systems,
College of Computing and Informatics, University of North
Carolina, Charlotte, NC, USA
e-mail: mshehab@uncc.edu

M. Ko
e-mail: mnko@uncc.edu

H. Touati
e-mail: htouati@uncc.edu

1 Introduction

Social network services, including Facebook, MySpace, Twitter, LinkedIn, Hi5, and Orkut have gained user adoption on the Internet over the past years. Different social network services provide users with different sets of services and experiences, for example, Facebook and MySpace allow users to create photo albums, fan clubs, and post feeds along with sharing all this content with friends, and LinkedIn enables users to connect with other users for professional purposes. To enjoy these services, users end up creating accounts on different sites and having multiple social network accounts, for example most Twitter users have a Facebook account (Branckaute 2010), and 64% of MySpace users have accounts in Facebook (Patriquin 2007). As users have multiple social network accounts, users start to connect social network accounts to interact with friends in different social network services. For instance, a user can connect his Twitter feed to his Facebook status such that his Facebook status will be updated automatically whenever he updates his Twitter feed (Schwartz 2010). Sharing content with friends in different social network services allows users to interact with friends across sites. Leading social network services are moving towards meeting the user's cross-site interactions demands that might include sharing photos, videos, and other content across sites (Gummelt 2010). However, current social network services do not support appropriate access control mechanisms for shared content across sites, which

Published online: 07 February 2012

makes users hesitant to securely share content across sites to avoid the risk of user privacy breaches. For social networking sites to provide a cross-site management framework would require consensus among the different sites on a unified user identity, object identifiers, policy language, and other cross platform semantics.

Enabling cross-site interactions beyond social network site boundaries is a challenging task that is related to both the semantics and the policies of the involved sites. In this paper, we propose a cross-site framework *x-mngr* for social network sites. The goal of this framework is the management of content sharing and access control across social network sites. We provide a cross-site access control policy, which enables users to specify policies that allow/deny access to their posted objects across social network sites. To enable cross-site interactions, the *x-mngr* should be able to mediate access between sites and map user's identities across social network sites. Cross-site identity mapping is required to enable *x-mngr* to easily identify friends that should be blocked and others that should be given access to shared content across social network sites. A simple solution would request the profile owner to match his/her friend's identities in different sites, which is a tedious and time-consuming task. For this purpose, we propose a partial mapping approach based on a supervised learning approach to recommend the blocked/allowed friends across sites. Our approach requires the profile owner to provide a small set of user's identity mappings across the different sites, which is used to generate the training set for the supervised learning algorithm. The training set leverages both profile attributes and network metrics for each user to capture their similarity across different sites. Furthermore, we explore the fusion of mapping decisions generated by user's neighboring friends or other trusted users in the social network to enhance the accuracy of the supervised learning approach. We implemented our proposed framework as a photo sharing application, which allows users to share photos between both Facebook and MySpace platforms. Our experiments indicate that our approach provides high accuracy in performing profile matches.

The rest of this paper is organized as follows. Section 2, describes current content sharing mechanisms in social network services and requirements. Section 3, provides an overview of social network, and the user profile model as a set of attributes and network metrics, and then discusses the access control model adopted by most social network frameworks. Section 4, gives an overview of our proposed *x-mngr* framework, describes the requirements for secure cross-site interactions, and discusses the partial matching problem. Section 5 discusses the supervised learning mechanism and describes the stages involved in requesting user's inputs, generating the training sets, training multiple classifiers, fine tuning the selected optimal classifier, and the classifier selection and fusion mechanisms used by the *x-mngr*. Section 6 describes the implementation details of our proposed framework. The experimental results are discussed in Sect. 7. The related work is discussed in Sects. 8, and 9 concludes the paper.

2 Background and challenges

Current social network sites provide immature content sharing mechanisms for outside of a single social network site. One solution is making a content public, where everyone is able to access the content from inside and outside the social network site. Another solution is sending a secret-link of the content to friends outside of the social network via email or other messaging tools. Recently, social network sites have introduced connecting user account services. For example, currently a user is able to connect Facebook and Twitter accounts to update and synchronize news feeds and account status between the two sites. These sharing solutions are not sufficient to users who want to share content across social network sites in a controlled manner. Making content in public is inadequate. A secret-link comes with security concerns, since friends can easily forward the secret link to others unauthorized users. To reduce this security concern, some sites request login to their sites to access the shared content using a secret-link, but it is inconvenient to users who do not have an account and they might not want to create new accounts in other sites just for accessing a shared content. As for the status/feed sharing, users only have opt-in/out choices across sites. It is not an expressive policy. There is a need for better mechanisms for secure content sharing across social network sites. To illustrate the challenges involved, we use the following scenario throughout this paper.

> Alice's high school friends and music club friends are mainly using SN_B, and her college friends and her work friends are using SN_A. To maintain online friendship with them Alice has accounts on social network site A (SN_A) and social network site B (SN_B). Her friends on two social network sites don't want to migrate or access other social network site, so Alice has uploaded a same content to both sites whenever she would like to share some content with them. One day, Alice want to share her wedding album with high school friends in SN_B and college friends in SN_A. However, she doesn't want to share the wedding album with her ex-boyfriend Bob.

From the above scenario, we formulate a set of core requirements for *x-mngr* as following:

- **R1**. The content owner should be able to share contents from one social network site (SN_B) to another social network site (SN_A) without the need to upload the same content to other social network site.

- **R2**. Friends on SN_A do not need to create accounts in SN_B to access the shared contents. Friends are able to access the shared content from their favorite social network site (SN_A).
- **R3**. The content owner should be able to compose access control policies for shared contents across sites. The content owner should be able to select groups or friends on other social network site (SN_A).
- **R4**. Shared content should not be accessed by unintended friends across sites. For example, if Alice blocks Bob from accessing the wedding album on SN_B, he must be blocked from accessing the shared wedding album on SN_A.

The *x-mngr* framework provides the requirements R1–R3 by providing a *cross-site* policy, that enables both the authoring and management of cross-site accesses. The last requirement (R4) is another important challenge, which is addressed in this paper. To prevent the unintended access across sites, a *complete* mapping mechanism requires the *x-mngr* to have a complete set of user friends' identity mappings between all similar users in both SN_A and SN_B. Requiring a complete set of user identity mappings is not realistic as it will require all users to explicitly and truthfully specify all their accounts in different sites. According to our exploration of existing identity mapping solutions, attribute based matching are inaccurate due to deception, errors, or missing attribute (Wang et al. 2005; Xu et al. 2007). Email based matching is only available when users use a same email across sites. All social network sites use a same federated identity management system is infeasible. To overcome these limitations, we investigate the adoption of a *supervised learning* technique.

3 Preliminaries

Users and relationships between users are the core components of social networks. Each user manages an online profile, which usually includes information such as the user's name, birth date, address, contact information, emails, education, interests, photos, music, videos, blogs, and many other items. Each user $u_i \in V$ maintains a profile, which is composed of N profile attributes, $\{A^i_1, \ldots, A^i_N\}$. Each attribute is a name-value pair (an, av), where *an* and *av* represent name and value, respectively. For example, a Facebook user profile includes attributes such as birthday, location, gender, religion, etc. Users are also able to post objects such as photos, videos, and statuses to their profiles to share with other users.

Users are connected to a set of friends, using this notion a social network can be modeled as an undirected graph $G(V, E)$, where the set of vertices V is the set of users, and the set of edges E is the set of friendship relationships between users. The edge $(u_i, u_j) \in E$ implies that users u_i and u_j are friends. Other forms of relationships between users exist, such as group memberships and user interactions; in this paper we only consider friendship relationships. Using the graph-based model for social networks, we leverage the node network structural properties to provide additional user attributes. These attributes include several small world network metrics such as node degree centrality, betweenness, hit rate, eigen values (Scott 2011; Borgatti and Everett 2006; Newman 2001). Each metric provides a different indicator about the user, for example the degree shows how popular is a user; Short and Hughes 2006 used the centrality measures of degree and betweenness to analyse relationships between street gangs members. Leaders, lurkers, associates, spammers, and influential users can be detected by studying the network metrics and network based clustering approaches (Fazeen et al. 2011; Gilbert et al. 2011; Saravanan et al. 2011). For a user u_i, we are able to compute M network metrics $B_i = \{B^i_1, \ldots, B^i_M\}$. Each metric provides a different indicator about users in a given social network (Newman 2003, 2001; Kleinberg 1999). Each user u_i in a social network maintains a collection of user profile attributes and a set of user friendships of which social network metrics are computed, $P_i = \{A_i, B_i\}$.

3.1 Access control in social networks

Users are able to post or download content objects such as photos, videos, and statuses on their profiles to share them with other users. A user u_i posting an object O on her profile is allowed to setup an access control policy to specify which friends are allowed/denied access to the posted object. The access control policy is managed and stored by the hosting social network site. We define an access control policy as

Definition 1 The access control policy of an object O is defined using two access control lists, namely the allow list ACL^+ and the exception list ACL^-, which are sets of the allowed and the denied users or groups, respectively. Access control follows the closed world assumption, where if access is not explicitly specified it is assumed to be not accessible. For an object O given ACL^+ and ACL^-, a user u is given access to O iff $u \in ACL^+$ and $u \notin ACL^-$, or in compact form $u \in (ACL^+ \setminus ACL^-)$.

For example, in SN_B the user Alice would like all her high school friends (Group G_1) and Music Club (Group G_2) to be able to access her posted wedding photo album ($AID = 12345$) except her friends Bob (user $U1$). Accordingly, for this photo album $ACL^+ = \{G_1, G_2\}$,

$ACL^- = \{U1\}$, and access is only given to users in $\{G_1, G_2\} \setminus \{U1\}$ refer to Fig. 1. This exception based group approach (ACL^+, ACL^-) is commonly adopted by state of the art social network sites such as Facebook.

4 Cross-site framework

Current social network architectures only allow users to compose localized policies that control access with respect to users subscribed on the site where the objects are posted. Users are not able to share the posted objects across social network sites in a control manner. We propose the *x-mngr* framework as shown in Fig. 1. We assume the *x-mngr* is a trusted party that will manage access across multiple social network sites. We design *x-mngr* to address all of the previously identified core requirements for the cross-site sharing in the Sect. 2. The *x-mngr* puts the object owners in full control of access to their posted objects across social network sites. The *x-mngr* controls access of posted objects relying on owner's *local policy* and *cross-site* policy for the posted objects. We refer to the site that hosts the content object as the *target* site, and the request is generated from the *viewer* site. The *cross-site* policy is defined as follows:

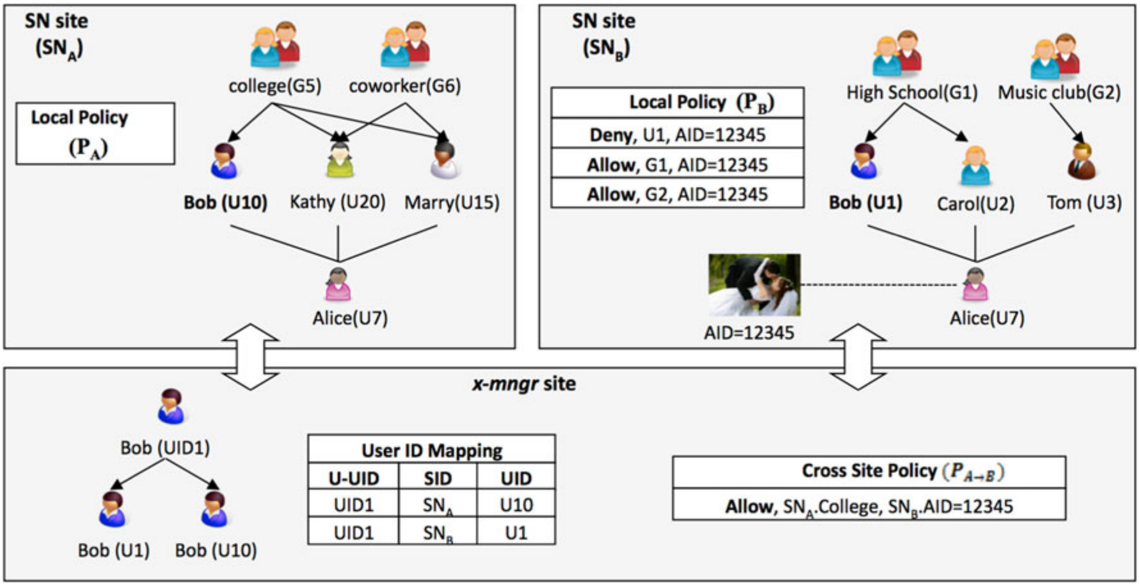
Definition 2 (*Cross-site policy*). Given a *viewer* site SN_A and a target site SN_B, the *cross-site* policy $P_{A \to B}$ specifies the access control lists (O, ACL^+, ACL^-) w.r.t subjects from the viewer site SN_A and objects from the target site SN_B.

In this setting, there are two classes of policies, namely the *local* policy and the *cross-site* policy. For example, Alice, which is the *focus* user, posted her wedding album photos in the target site SN_B. Alice also has an account in the viewer site SN_A and would like to share the album in site SN_B with her college friends (Group G_5) in site SN_A. The corresponding *cross-site* policy for Alice's wedding album is $P_{A \to B} = (O = \{SN_B.AID = 12345\}, ACL^+ = \{SN_A.G_5\}, ACL^- = \{\})$. Figure 1 shows sites SN_A and SN_B, the *local* policy in site SN_B and the *cross-site* policy manager maintaining the *cross-site* policy $P_{A \to B}$.

Enabling cross-site interactions should maintain both the autonomy and security principles of secure interoperation (Gong et al. 1996, 1994; Shehab et al. 2005). The autonomy principle requires that any access permitted within an individual site must also be permitted in the same site under secure interoperation, for example, if Tom was able to access Alice's photo album in SN_A before connecting with SN_B, the autonomy principle requires that Tom should still be able to access this album in SN_A after connecting to SN_A to SN_B. The security principle requires that any access not permitted within an individual site must also be denied under secure interoperation, for example, if Tom is not able to access Alice's photo album in SN_A, should still be denied access to this album after connecting SN_A to SN_B.

Definition 3 (*Safe*). A cross-site manager is *safe* if it does not deny legal requests or permit illegal requests from a viewer site to a target site.

The cross-site manager has no control on enforcing the *local* policy in local sites, for example, the *local* policy P_B is controlled and enforced by the site SN_B irrespective of the cross-site manager decisions. This implies that the *autonomy principle* is obeyed. The challenge is to enforce the *security principle*, as it requires the cross-site manager to deny access



Local Policy (P_B)
Deny, U1, AID=12345
Allow, G1, AID=12345
Allow, G2, AID=12345

User ID Mapping		
U-UID	SID	UID
UID1	SN_A	U10
UID1	SN_B	U1

Cross Site Policy ($P_{A \to B}$)
Allow, SN_A.College, SN_B.AID=12345

Fig. 1 *x-mngr* cross-site access framework

to objects that would have been denied by the target site's *local* policy P_B. For an object $O \in SN_B$ with a *local* policy P_B defined as $P_B.ACL^+$ and $P_B.ACL^-$, and a *cross-site* policy $P_{A\to B}$ defined by $P_{A\to B}.ACL^+$ and $P_{A\to B}.ACL^-$, a user u from the viewer site SN_A is given access to object $O \in SN_B$ if the all the below conditions are satisfied:

- **C1**. $u \in (P_{A\to B}.ACL^+ \setminus P_{A\to B}.ACL^-)$
- **C2**. $M_{A\to B}(u) \notin P_B.ACL^-$

The first condition (C1) ensures that the requesting user $u \in SN_A$ from the viewer site is permitted access via the *cross-site* policy $P_{A\to B}$. The second condition (C2) involves the user identity mapping function $M_{A\to B} : u \to v$, where $u \in SN_A$ and $v \in SN_B$, which maps a user u from a viewer site SN_A to a corresponding user v from the target site SN_B. The mapped user $v = M_{A\to B}(u)$ is checked against $P_B.ACL^-$ to ensure that this user is not explicitly denied access by being in the exception access list in the *target* site SN_B. Condition (C2), ensures that the exception list of the *target* site is respected and is not violated when requests are made through the *x-mngr* framework. In other words, the exception list of the local *target* policy overrides the cross site policy.

The *cross-site* policy $P_{A\to B}$, manages requests generated by users from the viewer SN_A to the target SN_B, the user identities in $P_{A\to B}$ are w.r.t user identities in SN_A, thus ensuring first condition (C1) is simply accomplished by executing the *cross-site* policy against the user identities in the viewer site SN_A. However, the second part is not trivial because $P_B.ACL^-$ is w.r.t the users identities in SN_B, and users can easily create accounts in different social network sites and have different relationships with the same person in different sites, which would require *x-mngr* to maintain a mapping function $M_{A\to B}$ between user identities in viewer and target sites. For example, in Fig. 1, user Bob has an account in sites SN_A and SN_B with user accounts $U10$ and $U1$, respectively. Bob is Alice's friend in both sites. Alice's wedding photos are posted in site SN_B and the *local* policy in SN_B will deny Bob access to this album in SN_B. However, the *cross-site* policy $P_{A\to B}$ enables all Alice's college friends access to her photo album from viewer site SN_A. If the *x-mngr* has a mapping between friends' identities in SN_A and SN_B, then it can compute $M_{A\to B}(U1)$ to find out that Bob is $U10$ in site SN_B and block his access from SN_A. As we discussed in Sect. 2, we adopted supervised learning technique to map friends' identities across social network sites. The detail of our approach is discussed in the following sections.

5 Supervised learning and x-mngr

In machine learning literature, a learning model is a function f that takes as an input a set of attributes and returns a label or classification. For example, a function that takes the user's age, sex, credit rating, and job status and generates a recommendation to either grant a loan or no. A supervised learning mechanism uses previous cases or training data Θ to learn the function f, which we refer to as f_Θ.

Taking a simple user-centric approach to address the profile-matching problem would require that each focus user (content owner) manually provide mappings between all similar profiles of his friends on different social network sites. Usually, this is a tedious task and the user will end up ignoring this task. Furthermore, while users can limit access of their profiles via privacy setting, user's perceptions of visibility do not always match with reality (Boyd 2008), let alone managing cross-site policies. Instead, the approach we adopt is an assisted user centric approach, where the focus user is required to only provide an initial small subset (α) of the profile mappings which are used as examples (training set Θ) to train a supervised learning algorithm that can classify mappings between user profiles. Basically, we attempt to learn the mapping function $f_\Theta : \mathcal{X} \to \mathcal{Y}$, where

1. $\mathcal{X}$ is a set of attributes describing the profile difference vector (discussed in the next subsection).
2. $\mathcal{Y}$ is a set of labels $\{y_0, \ldots, y_m\}$, in our case it is $\{match, no - match\}$, representing match or no-match, respectively.
3. Θ is the training set, which is a set of composed of example matching and no-matching friends' profile pairs.

Our goal is to learn the function f_Θ based on the provided data set Θ. Once f_Θ is learned, we can automatically decide if a given pair of user profiles $P_i \in SN_A$ and $P_j \in SN_B$ are owned by the same user or not. This learning mechanism is a supervised learning (Liu 2007) as it requires an example data set to train and guide the generation of the mapping function f_Θ. Given a pair of friends u_p and u_q belonging to the social network SN_A and $SN_{B,}$ respectively, the classifier f_{Θ_i} for user u_i assigns the label y_l to this user pair (u_p, u_q) provided that this label maximizes the classifier's confidence or probability measure $P((u_p, u_q) \to y_l|\Theta_i)$ based on the training set Θ_i. For more information about supervised learning algorithms the interested reader is referred to (Liu 2007; Witten et al. 2005). Supervised learning is used because it allows users to guide the classifier by providing the initial training set; this approach ensures the users can easily customize their mapping function. This class of learning has proven its effectiveness in the context of policy management to enable users to automatically setup their social network policies (Fang and LeFevre 2010), and identify trusted and non-trusted users (Shehab et al. 2010).

The steps involved in the learning-based profile matching process are described in Fig. 2. Step 1 is a data collection stage in which the *x-mngr* retrieves the focus user friends' profile and network attributes from sites SN_A and SN_B. The collected user profiles might be missing some attributes, we use heuristics to estimate such missing attributes, for example a user's missing age could be estimated as the average of all their friends' ages (Wasserman and Faust 1994). In step 2, the *x-mngr* presents the focus user with her friends from SN_A, SN_B and requests the user to indicate at least α users in both sites. A mapping between user $u_p \in SN_A$ and user $u_q \in SN_B$ is the pair (u_p, u_q), indicating that user u_p and u_q belong on the same user. A training set is generated using all the α mapping pairs (u_p, u_q). In step 3, the generated training set Θ can be used to directly train a classifier. However, there are several classifiers algorithms and it is crucial to select classifier that is most suited for this specific user instance. So, the mechanism we adopt is to train and tune several classifiers and then compare their performance based on standard cross-validation methods such as n-fold cross validation (Witten and Frank 2005). Given m classifiers $\{f_\Theta^1, \ldots, f_\Theta^k\}$, the classifier with the lowest error rate is selected, which is denoted as f_Θ^*.

In step 4, the knowledge accumulated by other users in the social network can be utilized to further enhance the classifier accuracy. It is important in this step to seek classification advice from other users who are able to map users similar to the focus user. This is referred to as the selection process where β other user classifiers are selected based on their accuracy in labeling the focus user's training set. The decisions of the selected β classifiers are fused with the focus user's classifier to generate the focus user's mapping function $\tilde{M}_{A \to B}$.

Finally, in step 5, the selected mapping function $\tilde{M}_{A \to B}$, is used to decide if a user from site SN_A maps to a user in the target site SN_B *local* policy exception list $P_B.ACL^-$. The details of this approach are discussed in the following section.

5.1 Training set generation

Given two users $u_i \in SN_A$ and $u_j \in SN_B$, with profile attributes and network metrics $\{A_i, B_i\}$ and $\{A_j, B_j\}$ respectively, we define the distance vector as follows:

$$\begin{aligned} D(i,j) &= [d(A_i, A_j), d(B_i, B_j)] \\ &= [d(a_i^1, a_j^1), \ldots, d(a_i^N, a_j^N), d(b_i^1, b_j^1), \ldots, d(b_i^M, b_j^M)] \end{aligned}$$

The distance function $d(.,.) \in \mathbb{R}^+$ is dependent on the data attribute domain, where $d(a,a) = 0$. These distance

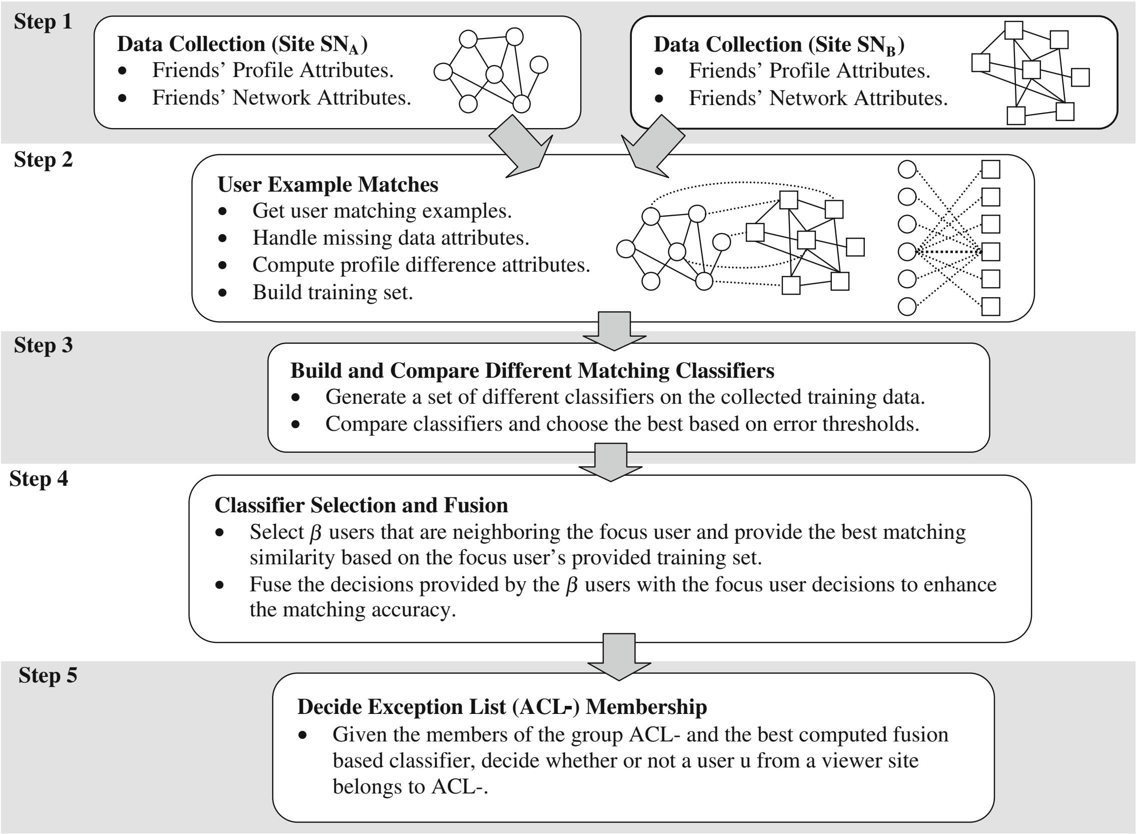

Step 1
Data Collection (Site SN_A)
• Friends' Profile Attributes.
• Friends' Network Attributes.
Data Collection (Site SN_B)
• Friends' Profile Attributes.
• Friends' Network Attributes.
Step 2
User Example Matches
• Get user matching examples.
• Handle missing data attributes.
• Compute profile difference attributes.
• Build training set.
Step 3
Build and Compare Different Matching Classifiers
• Generate a set of different classifiers on the collected training data.
• Compare classifiers and choose the best based on error thresholds.
Step 4
Classifier Selection and Fusion
• Select β users that are neighboring the focus user and provide the best matching similarity based on the focus user's provided training set.
• Fuse the decisions provided by the β users with the focus user decisions to enhance the matching accuracy.
Step 5
Decide Exception List (ACL-) Membership
• Given the members of the group ACL- and the best computed fusion based classifier, decide whether or not a user u from a viewer site belongs to ACL-.

Fig. 2 Steps in generating the user centric match classifier

values of each profile attribute and network attribute are considered together in classification process to decide the matched profile. Assume the focus user has R and S friends in SN_A and $SN_{B,}$ respectively, with a user-mapping (i, j) this provides $R + S - 1$ classified mappings, namely

$$\begin{aligned} &\text{``Match''} && (u_i, u_j) \\ &\text{``No-Match''} && \forall(u_i, u_s) \text{ where } u_s \in SN_B \wedge u_s \neq u_j \\ &\text{``No-Match''} && \forall(u_r, u_j) \text{ where } u_r \in SN_A \wedge u_r \neq u_i \end{aligned}$$

By explicitly indicating the match (u_i, u_j), the focus user is implicitly indicating that user u_i is not same to all other friends in SN_B and similarly user u_j is not same to all other users in SN_A. The distance vector is computed for both the explicit match and implicit no-matches and then used as the training set Θ.

5.2 Attributes and network distances

In order to measure the similarity value of each attribute pair, we consider different similarity methods for different attribute types. In case of string attributes such as school name, last name, and first name, these attributes are tokenized and normalized before computing the distances. Through the tokenization process, the string attribute value is divided into tokens by converting a sequence of characters into a sequence of tokens. For example, if the attribute value of school name is “UNC Charlotte”, it generates (“UNC”, “Charlotte”) as tokens. Then the normalization process, the process of canonicalizing token, matches the semantically equivalent token despite superficial differences in the character sequences. For instance, “UNC” and “University of North Carolina” should be considered as the matched token. In case of attribute value of first name, “joe” and “Joseph” is also considered as the matched token via normalization process. This normalization process increases the accuracy of similarity score for the different format of string attributes.

After the tokenization and normalization, we apply the Levenshtein Distance (Levenshtein 1966), a metric for measuring the amount of difference between two strings attributes. For numeric attributes such as age we use the square Euclidian distance. For address attributes, we first perform the geocoding process of converting the addresses into their geographic coordinates represented as latitude and longitude; then we compute the distance between the two geocoded addresses (http://code.google.com/apis/maps/).

The network metrics are numeric attributes. When comparing metrics computed from different graphs the varying size of the graphs presents a challenge. For example, a user in Facebook might have 300 friends while having in MySpace only 100 friends; due to the different network sizes the metrics computed will differ considerably. To enable the comparison of metrics computed from different graphs we adopt the approach presented by Brandes and Erlebach (2005), which normalizes each network metric based on a specific normalization factor. Then the Euclidian distance is used to compute the distance between the normalized metrics from different social networks.

5.3 Classifier selection and fusion

The inherent advantage of social networks is the ease of sharing of news, photos, videos, and several other data objects among users. We extend this sharing to include the accommodation of user experiences by leveraging their trained match classifiers, where user u_j is able to share their matching function f_{Θ_j} with other users. Assume a user u_i would like to leverage the experience of other users in the social network to improve their matching function f_{Θ_i}. In this section, we use f_{Θ_k} to refer to the best match classifier $f^*_{\Theta_k}$ for user u_k. Given a user u_i and a set of users $S = \{u_1, \ldots, u_n\}$, the set S can be chosen from the neighboring trusted friends or from other experienced users in the social network. Each user u_k in the set S is willing to share their matching function f_{Θ_k} to improve the matching function of user u_i. As indicated in Fig. 3, this translates into two substeps: (1) the selection of β users from the set S that are best fit to help user u_i in computing an improved matching function and (2) the fusion of the different f_{Θ_k} functions provided by the β users with the focus user's function f_{Θ_i}.

Definition 4 (*Selection*) Given a user u_i, a set of user trained classifier functions $f_S = \{f_{\Theta_1}, \ldots, f_{\Theta_n}\}$, the training set Θ_i for user u_i, and a classifier fitness function $\Phi : f_{\Theta_k} \times \Theta_i \rightarrow \Re$, select the best β classifiers based on the fitness function.

The selection process is based on the fitness function as defined in Def. 4. The fitness function is a mechanism to

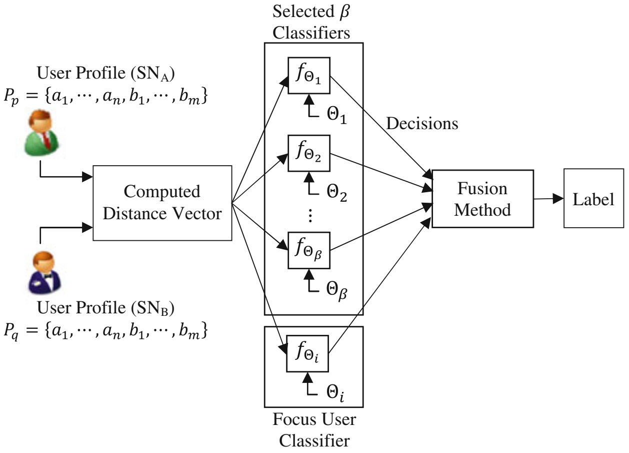


Fig. 3 Classifier selection and fusion

rank the classifiers in f_S based on their similarity to the decisions taken by the classifier of user u_i. The fitness function tests each classifier f_{Θ_k} by labeling the tuples in the training set Θ_i and computing the vector $[TP, TN, FP, FN]^T$, where TP = True Positive, TN = True Negative, FP = False Positive, and FN = False Negative. The fitness of f_{Θ_k} is based on the classifier accuracy of recall and precision (Perols et al. 2009; Barbosa and Freire 2007). The β classifiers with the highest fitness are selected and are denoted by the set $S_\beta = \{f_{\Theta_1}, \ldots, f_{\Theta_\beta}\}$.

Given the β classifiers, the next step involves fusing the decisions of these classifiers and the decisions generated by the focus user's classifier (f_{Θ_i}) to improve the classification result. We adopt the most relevant classifier fusion algorithms (Kittler et al. 1998): *group voting*, *group confidence product,* and *most confident*. The group voting mechanism is based on selecting the label (e.g., match or not-match) which receives the largest number of votes by the β classifiers. Given a user pair (u_p, u_q), where $u_p \in SN_A$ and $u_q \in SN_B$, the label w_l is assigned to this user pair if w_l receives votes as follows:

$$\sum_{k=1}^{\beta} \delta_k^l(u_p, u_q) = \max_{r=1,\ldots,m} \sum_{k=1}^{\beta} \delta_k^r(u_p, u_q)$$

where

$$\delta_k^r(u_p, u_q) = \begin{cases} 1 & \text{if } f_{\Theta_k}(u_p, u_q) = w_r \\ 0 & \textit{otherwise} \end{cases}$$

The group confidence product mechanism is based on selecting the label that maximizes the product of the confidence of all the β classifiers. For a user pair (u_p, u_q), the label w_l is selected if group confidence product of w_l is as follows:

$$\Pi_{k=1}^{\beta} P((u_p, u_q) \rightarrow w_l | \Theta_k) = \max_{r=1,\ldots,m} \Pi_{k=1}^{\beta} P((u_p, u_q) \rightarrow w_r | \Theta_k)$$

The most confident mechanism is based on selecting the class that gets the highest confidence from any of the β classifiers. This approach fuses the different classifier confidence and adopts only the label provided by the most confident classifier. For a user pair (u_p, u_q), the label w_l is selected if the confidence of w_l is as follows:

$$\max_{k=1,\ldots,\beta} P((u_p, u_q) \rightarrow w_l | \Theta_k) = \max_{k=1,\ldots,\beta} \max_{r=1,\ldots,m} P((u_p, u_q) \rightarrow w_r | \Theta_k)$$

After β classifiers with the highest fitness are selected, an appropriate fusion algorithm (of the three listed above) is chosen to fuse the results of the f_{Θ_k} functions producing a predicted label, i.e., match or no-match. This final fused classifier represents the identity mapping function $\tilde{M}_{A \rightarrow B}$ between users in site SN_A and SN_B.

5.4 Cross-site mechanism

After defining the *cross-site* policy $P_{A \rightarrow B}$ and learning the mapping function $\tilde{M}_{A \rightarrow B}$, the *x-mngr* will integrate the two solutions into the framework and apply them to resolve requests made by users from the viewer site SN_A to access the target site SN_B. Given a *local* policy P_B for site SN_B defined as $P_B.ACL^+$ and $P_B.ACL^-$, a *cross-site* policy $P_{A \rightarrow B}$ defined by $P_{A \rightarrow B}.ACL^+$ and $P_{A \rightarrow B}.ACL^-$, the learned mapping function $\tilde{M}_{A \rightarrow B}$, a request from user $u_q \in SN_A$ to access resources in site SN_B is permitted if the user is allowed access through the *cross-site* policy $P_{A \rightarrow B}$ and the mapping function $\tilde{M}_{A \rightarrow B}$ does not map the user u_q to any user $u_p \in P_B.ACL^-$. The accuracy of this approach are closely related the accuracy of the mapping function $\tilde{M}_{A \rightarrow B}$. The probability of error of this approach (false positive and false negative) can be summarized as

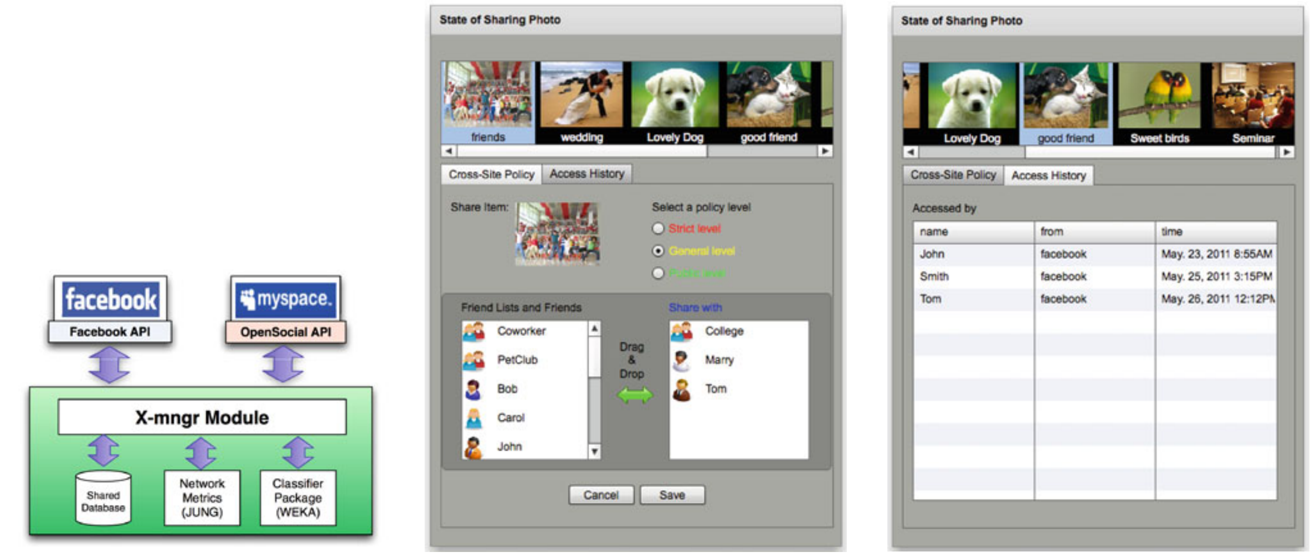
$$P_e = P(\tilde{M}_{A \rightarrow B}(u_q) \notin P_B.ACL^-, \text{Deny}) + P(\tilde{M}_{A \rightarrow B}(u_q) \in P_B.ACL^-, \text{Allow})$$

The first and second components of P_e can be computed as $P(\tilde{M}_{A \rightarrow B}(u_q) \notin P_B.ACL^- | \text{Deny}) P(\text{Deny})$ and $P(\tilde{M}_{A \rightarrow B}(u_q) \in P_B.ACL^- | \text{Allow}) P(\text{Allow})$ respectively. Note, the above probability of error can be derived from the probability of true negative and false positive of the mapping function $\tilde{M}_{A \rightarrow B}$. We investigate the accuracy and precision of this approach in the Sect. 7.

6 Implementation details

As a proof of concept and for data collection, we provided an implementation of our proposed framework to enable secure interactions between users across social network sites such as Facebook and MySpace. We developed a *crossAlbum* application that enables users to share photo albums with their friends across social network sites. Our crossAlbum application uses to Facebook Connect and MySpaceID to authenticate users in both Facebook and MySpace, respectively. Using the Facebook and MySpace API framework enables our application to request user authorization to access their profile. In addition, we presented the users with a consent form highlighting the data collection and aggregation practices adopted.[1] The Facebook API and OpenSocial API were used to enable our crossAlbum application of retrieving the users' profile,

[1] IRB Protocol No: 09-03-16, Title: Cross-site Interaction between Social Networks.



State of Sharing Photo

Cross-Site Policy | Access History

Share Item:

Select a policy level
○ Strict level
● General level
○ Public level

Friend Lists and Friends: Coworker, PetClub, Bob, Carol, John

Share with: College, Marry, Tom

State of Sharing Photo

Cross-Site Policy | Access History

Accessed by

name	from	time
John	facebook	May. 23, 2011 8:55AM
Smith	facebook	May. 25, 2011 3:15PM
Tom	facebook	May. 26, 2011 12:12PM

(a) *crossAlbum* application modules **(b)** The Cross-Site Policy panel **(c)** The Access History panel

Fig. 4 crossAlbum implementation

friend list, and friend information in both Facebook and MySpace. Figure 4(a) describes the main modules of our implementation. The shared database is a MYSQL database storing user's identity map, cross-site policies, and shared photo's address. The network metrics for different users are computed using the open source Java Universal Network Framework (JUNG) (The JUNG Framework Development Team: Java Universal Network/Graph Framework. http://jung.sourceforge.net/ 2009). The classifiers were generated using the open source Java WEKA library (The University of Waikato: WEKA Machine Learning Project. http://www.cs.waikato.ac.nz/ml/index.html 2009).

To target users who have accounts in both social networks we invited users to add our *crossAlbum* application by posting invitation messages on Facebook (MySpace) groups in MySpace (Facebook). Once the user adds the *crossAlbum* application and authenticates in both social networks successfully, our application retrieves the participant's profile attributes, friendship connections, friend's attributes, and friend of friend information in both social networks. Upon completing this stage, the user is presented with pairs of recommended friend mappings computed base on matching first and last names of their friends in both sites, and the user is asked to confirm the correct recommended friends' mappings. In the following step is the manual friend mapping stage, the user is presented with a list of their friends in Facebook and is asked to indicate if they map to one of their friends in MySpace. This was implemented using the Adobe Flex to enable the user to easily map users by typing a few characters of the friend's name in a text box placed beside each friend profile photo. The collected friend mappings are used to generate distance values between friends in both sites using the collected profile and network attributes; the training set was generated and used to compose the classifier.

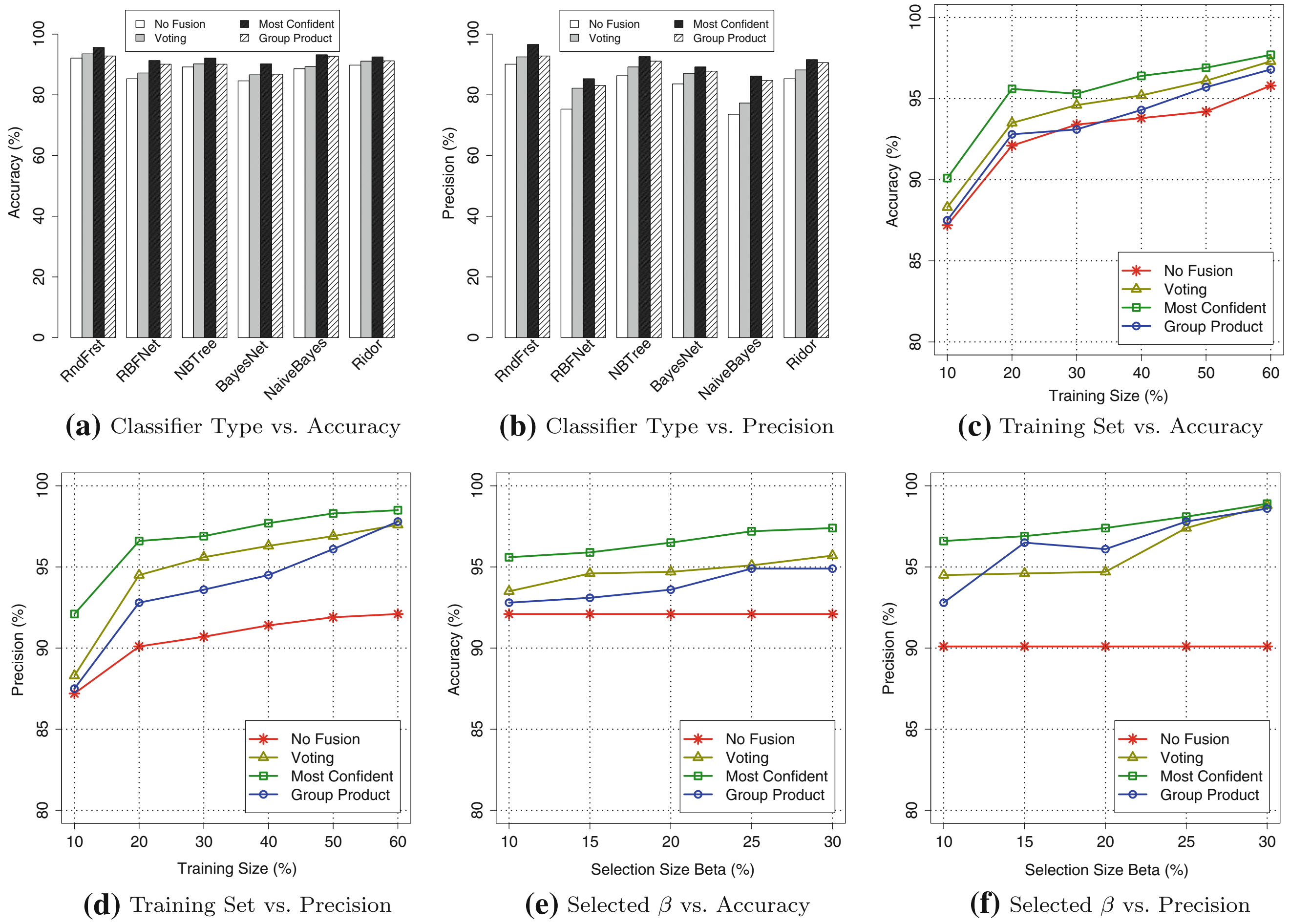
The *crossAlbum* application enables the users to add and view photos in one social network site and enable users who install the application to view it in other social network sites according to the *cross-site* policy. Our prototype *crossAlbum* has several options such as *add photos*, *view photos*, *setup policy* and *manage mappings*. The *add* and *view* options enable a user to add and view photos to the *crossAlbum* application. Upon adding a photo, the owner's uid, public address of photo, social network sites, and access control information are stored in a shared database.

The *setup policy* enables the user to share photos with other users in other social network sites and to setup the corresponding *cross-site* policy. Figure 4(b) shows a screen shot of the setup policy, where the user is able to select social networks sites and friend relationship to allow access in different sites. The application also retrieves the Facebook groups and friendslists to enable the users to specify a cross-site policy based on groups and individual users, see Fig. 4(b). The user is able to retrieve the list of users who are able to access the shared content in both sites. The application also provides the user with access history log summarizing the users who have access the shared content and the viewer/target details Fig. 4(c). Users are able to share content with users across sites only if they setup their cross-site policy. The *manage mappings* enables the user to inspect and specify training sets.

7 Experimental results

In order to investigate the effectiveness of the proposed user-mapping approach, we performed an extensive experimental evaluation on the collected data. We investigated the performance of different classifiers by varying training set size and for the different fusion schemes discussed in previous sections. In order to collect data, we invited 5,000 online users who have accounts in both Facebook and MySpace to install our *crossAlbum* application. Users were invited using Facebook and MySpace invitation messages requesting from users to install the application on their profile and to complete the user study. We did not provide any financial motivation or reward for users to complete the user study. The application installation, registration, and profile mappings were successfully completed by 193 users. The user sample only reflects users who maintained an account in both Facebook and MySpace. The users' profiles, friend's list, friend of friend's lists, and profiles were collected. The users were required to provide mappings between their friends in different sites. We collected 6,983 Facebook profiles, 9,973 MySpace profiles, and 1,378 profiles matches from the participants. For each user, we accumulated the profile attributes and computed the network metrics. The following profile attributes that were obtained were First Name, Last Name, Gender, Location, Date of Birth, and Education. In addition, each user's social graph was built and a series of network metrics were computed which include, degree, HUBS, authority, betweenness, closeness, PageRank, Eigenvector, and number of common friends.

The collected data was used to train six classifiers namely, BayesNet, NaiveBayes, NBTree, RandomForest, RBFNetwork, and Ridor. The true positive, true negative, false positive, and false negatives for each classifier were recorded. Figure 5(a, b), shows the accuracy and precision results generated by *x-mngr* for a training set of $\alpha = 20\%$, for the different classifiers and 10 friends selected for fusion ($\beta = 10$). Referring to Fig. 5(a, b), the Fusion schemes consistently provided higher accuracy and



(a) Classifier Type vs. Accuracy

(b) Classifier Type vs. Precision

(c) Training Set vs. Accuracy

(d) Training Set vs. Precision

(e) Selected β vs. Accuracy

(f) Selected β vs. Precision

Fig. 5 User-mapping experimental results

precision when compared with the no fusion scheme. Using the Random Forest classifier the accuracy was 92 and 97% for the no-fusion and most confident fusion approaches, respectively. The Random Forest classifier precision was 91 and 97% for the no-fusion and most confident fusion approaches, respectively. Based on the presented results, the proposed mapping approach presents high accuracy and precision for mapping users across social networks, which can be further improved by using adopting fusion based approaches.

Figure 5(c, d), presents and experiments conducted using the Random Forest classifier while varying the training set α from 10 up to 60%. The figures report the computed accuracy and precisions for different training set sizes. As expected, the classifier accuracy and precision increase as more user-mappings are provided for training. Note that the proposed fusion based approach provides higher accuracy and precision results when compared with the no-fusion approach over all training set sizes. For a training set of only 10% the most confident fusion based approach maintains an accuracy and precision of 90.1 and 92.5%, respectively, while the no-fusion approach maintained an accuracy and precision of 87.1 and 87.1%, respectively. The no-fusion mechanism provides high accuracy and precision results, in addition if the user is willing to provide the computing and memory resources we are able to provide higher accuracy and precision results using the fusion-based scheme. To investigate effect of the size of the selected fusion classifiers (β) we conducted experiments holding all parameters constant (Random Forest classifier, $\alpha = 20\%$) while varying β. Figure 5(e, f) depicts the accuracy and precision of the fused classifiers and the no-fusion classifier for different β (10–30) values. Note that as the number of fused classifiers β increase the fusion-based scheme provides higher accuracy and precision.

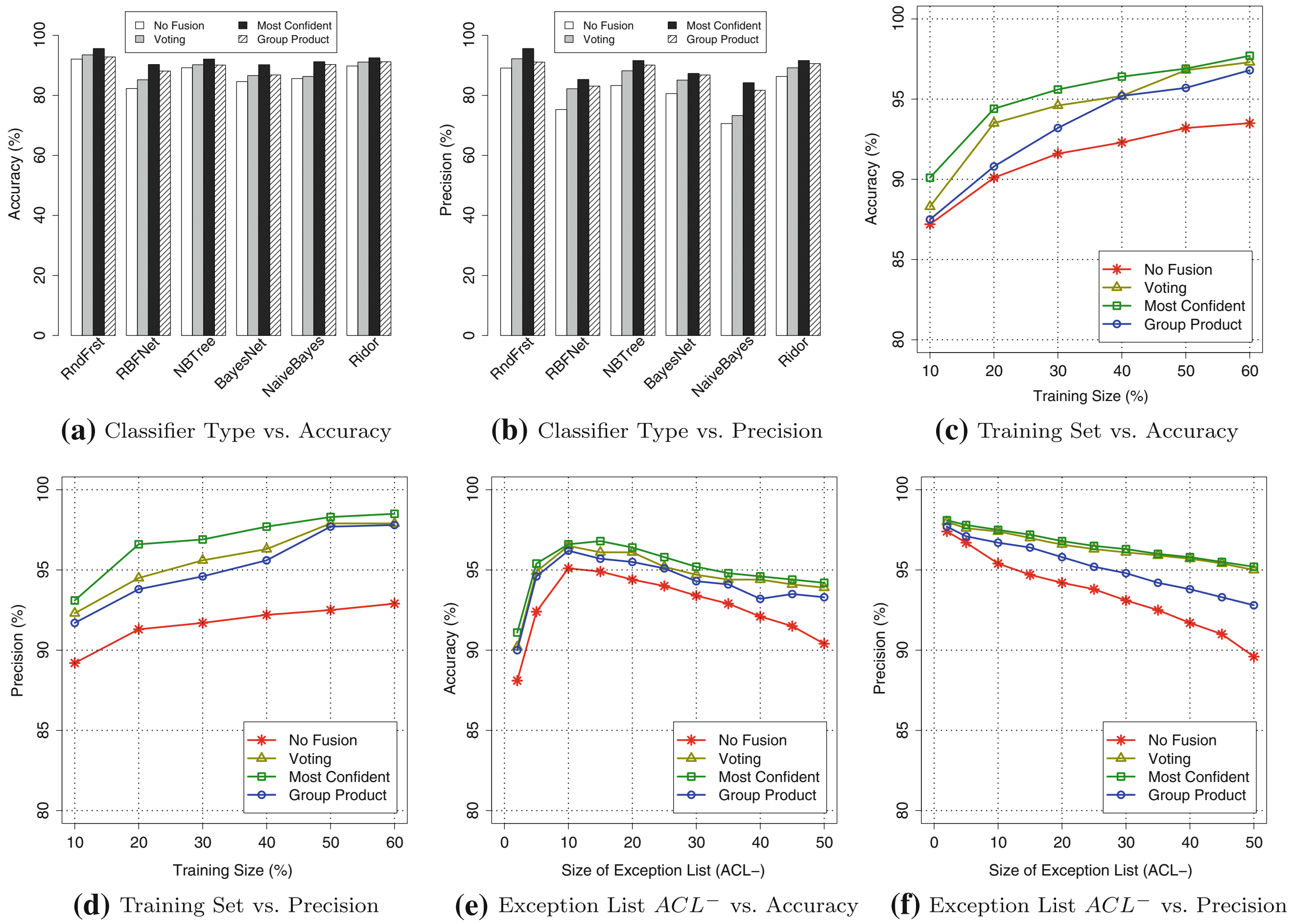
The effectiveness of *x-mngr* is dependent on whether users are given the correct set access permissions in the *cross-site* policy, and whether they are correctly identified in the target social site *local* policy exception list. In order to investigate the effect of the exception list ACL^{-} on the mapping process, we randomly generated different ACL^{-} sets and tested different mapping functions based on the trained classifiers. The experiments were repeated multiple times and averaged over all runs. Figure 6(a, b) shows the accuracy and precision results obtained while using all the classifiers, fixing the training set to $\alpha = 20\%$, and size of the ACL^{-} list to 2, which represents small exception lists. Figure 6(a, b) depicts the results of this experiment. The fusion-based approach consistently provided higher accuracy and precision when compared with the no-fusion approach for the different classifiers reported. The most confident fusion-based Random Forest classifier provided an accuracy and precision of 95.6 and 95.4%, respectively, in detecting users in the ACL^{-} set, while the no-fusion approach provided an accuracy and precision of 92.1% and 89.1%, respectively. Figure 6(c, d) presents the accuracy and precision for the Random Forest classifier while varying the training set size α (10–60%) and fixing size of ACL^{-} set to 2. The results show that the fusion approach performs better than the no-fusion approach. Both approaches showed an increase in accuracy and precision as the training size was increased. The fusion-based approach maintained a higher accuracy and precision compared with the no-fusion approach; the difference is about 5–10% enhancement in both accuracy and precision.

To investigate effect of the exception list size $|ACL^{-}|$ on the accuracy and precision of the our approach, we conducted experiments holding all parameters constant (Random Forest classifier, $\alpha = 20\%$) while varying $|ACL^{-}|$. Figure 6(e, f) depicts the accuracy and precision of the fused classifiers and the best classifier of the focus user (no fusion) for the different $|ACL^{-}|$ values (2–50). Note that as we increase ACL^{-} the accuracy and precision drop; this is because as the size of ACL^{-} increases there is a higher probability of false matches which affects both the accuracy and precision. Note that even though the accuracy and precision drops as $|ACL^{-}|$ increases the fusion-based classifier still consistently performs better than the no fusion classifier and maintains a less steeper decent in accuracy and precision. Furthermore, our fusion-based approach still maintains an accuracy of 94.5% and precision of 95.1% for an ACL^{-} of size 50.

The presented experimental results demonstrated the high accuracy and precision attained by our supervised base approach in both user-mapping and *cross-site* policy management. Our supervised learning approach shows an accuracy of 96.5% and a precision of 97.1% with a training set of 20% from the mapped friends, which demonstrates the applicability and suitability of our cross-site framework *x-mngr* for effectively mapping users across multiple sites and ultimately enabling secure cross-site interaction between different social networks.

8 Related work

Access control is one of the most important topics in social networks to protect user's privacy in sharing content with other people. Gollu et al. (2007) presented a social-networking based access control scheme suitable for online sharing of personal media. The authors consider user's identities as key pair and social relationship on the basis of social attestations. Access control lists are employed to define the access lists of users. Carminati et al. (2006) proposed a more sophisticated rule-based access control



(a) Classifier Type vs. Accuracy

(b) Classifier Type vs. Precision

(c) Training Set vs. Accuracy

(d) Training Set vs. Precision

(e) Exception List ACL^- vs. Accuracy

(f) Exception List ACL^- vs. Precision

Fig. 6 Exception list experimental results

model for social networks. It enforced complex policies expressed as constraints on the type, depth, and trust level of existing relationships. The authors also proposed using certificates for granting relationships authenticity, and the client-side enforcement of access control according to a rule-based approach, where a subject requesting to access an object must demonstrate that it has the rights of doing that. These papers focused on access control within a single social network but did not consider the access control for sharing content with outside of social networks.

Without a common identity management system between different sites, identity-matching techniques are used to detect the same user in different sites by utilizing user's information. Wang et al. (2004) proposed a record comparison algorithm that detects deceptive criminal identities using four personal attributes: name, date of birth, social security number, and address. It calculates the overall similarity score of personal attributes. If the overall similarity score is higher than a pre-defined threshold, two people are considered a matched people. Wang et al. (2005) revealed that incomplete records with many missing data could significantly increase the error rate of the record comparison algorithm that is a common limitation of many identity matching techniques using only personal attributes. Xu et al. (2007) showed that combining social features with personal features could improve the performance of criminal identity matching. They artificially constructed incomplete datasets from complete datasets by randomly choosing a percentage of person's records and removing their data of birthday or address values. Using this incomplete dataset with a decision tree classification method, they found out if the dataset had more missing values in personal identity attribute, the social contextual features significantly crease the matching performance. These papers showed how personal attribute and social features could affect the performance of the identity matching techniques but they did not investigate how network metrics can influence to the identity mapping. The identity matching techniques are based on the machine learning.

The machine learning has been used in various computer security areas to detect and prevent malicious activities and attacks more efficiently and reliably. From spam filtering to malware detection, machine learning becomes an essential component of computer security. Bratko et al. (2006) and Fumera et al. (2006) apply machine learning algorithms to classify the spam email. Barreno et al. (2008) applies the machine learning to privacy policy setting. From user's feedback about current policy, the machine learns incremental policy improvements and generates the refined policy. Maes et al. (1993) proposed using machine learning to automatically detect credit card fraud. Sinclair et al. (1999) leveraged machine learning in detecting anomalous packets on a data network. Ni et al. (2009) introduced the concept of using machine learning techniques for role base provisioning in role-based access control (RBAC) systems. Ni et al. describe how they use classifiers as a means to assign entitlements to roles. Our approach leverages classifiers in a similar fashion as a means to estimate identity matches across social networks based on both profile attributes and network metrics and by leveraging the fusion of other users' classifiers in the social network.

9 Conclusion

In this paper, we presented a new cross-site interaction framework that manages resource sharing and access control across social network sites. We provided a cross-site access control policy, which enables users to specify policies that allow/deny access to their posted objects across social network sites. We also proposed a partial mapping approach based on a supervised learning mechanism to map user's identities across social networks. We provide mechanisms to enable users to fuse other neighboring friends' mapping functions and to further enhance their mapping functions. Moreover, we demonstrated the feasibility of our framework by implementing a photo sharing application that allows users to share photos across Facebook and MySpace. Our experimental results show that our approach provides both high accuracy and precision in performing profile and exception list matching.

In the future, we will extend this framework to support the various cross interaction cases between different sites and investigate privacy and security issues related to such cross interaction. We will extend the cross-site access control policy as open standard to encourage the adoption of our framework by current social network sites. In addition, we will explore using ontologies to compute distances between profile attributes, and using other classification approaches such as fuzzy logic, and semi-supervised learning to assist in composing cross site profile mappings and policies. Furthermore, in this paper we assume sites have the same purpose and for future work we will investigate composing cross-site policies between sites with different purposes.

Acknowledgments This work was partially funded by the National Science Foundation (NSF-CNS-0831360 and NSF-CNS-1117411).

References

Barbosa L, Freire J (2007) Combining classifiers to identify online databases. In: WWW '07: Proceedings of the 16th international conference on World Wide Web. ACM, New York, pp 431–440

Barreno M, Bartlett PL, Chi FJ, Joseph AD, Nelson B, Rubinstein BI, Saini U, Tygar JD (2008) Open problems in the security of learning. In: AISec '08: Proceedings of the 1st ACM workshop on Workshop on AISec. ACM, New York, pp 19–26

Borgatti SP, Everett MG (2006) A graph-theoretic perspective on centrality. Soc Netw 28(4):466–484

Boyd D (2008) Taken out of context: American teen sociality in networked publics. Phd dissertation, University of California-Berkeley, School of Information

Branckaute F (2010) Twitter's meteoric rise compared to facebook [infographic]. The Blog Herald

Brandes U, Erlebach T (2005) Network analysis: methodological foundations, 1st edn. Springer, Berlin, Heidelberg

Bratko A, Filipič B, Cormack GV, Lynam TR, Zupan B (2006) Spam filtering using statistical data compression models. J Mach Learn Res 7:2673–2698

Carminati B, Ferrari E, Perego A (2006) Rule-based access control for social networks. In: On the move to meaningful internet systems 2006: OTM 2006 Workshops, vol 4278, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp 1734–1744

Fang L, LeFevre K (2010) Privacy wizards for social networking sites. In: Proceedings of the 19th international conference on World wide web, WWW '10. ACM, New York, pp 351–360

Fazeen M, Dantu R, Guturu P (2011) Identification of leaders, lurkers, associates and spammers in a social network: context-dependent and context-independent approaches. Soc Netw Anal Min 1:241–254

Fumera G, Pillai I, Roli F (2006) Spam filtering based on the analysis of text information embedded into images. J Mach Learn Res 7: 2699–2720

Gilbert F, Simonetto P, Zaidi F, Jourdan F, Bourqui R (2011) Communities and hierarchical structures in dynamic social networks: analysis and visualization. Soc Netw Anal Min 1:83–95

Gollu KK, Saroiu S, Wolman A (2007) A social networking-based access control scheme for personal content. In: Proceedings of 21st ACM symposium on operating systems principles (SOSP '07). Work in progress

Gong L, Qian X (1994) The complexity and composability of secure interoperation. In: SP '94: Proceedings of IEEE symposium on security and privacy. IEEE Computer Society, pp 190–200

Gong L, Qian X (1996) Computational issues in secure interoperation. IEEE Transact Softw Eng 22(1):43–52

Google Inc. (2009) Google Maps API Services. http://code.google.com/apis/maps/

Gummelt M (2010) Publishing to twitter from facebook pages, http://blog.facebook.com/blog.php?post=123006872130

Kittler J, Hatef M, Duin RP, Matas J (1998) On combining classifiers. IEEE Transact Pattern Anal Mach Intelligence 20(3):226–239

Kleinberg JM (1999) Authoritative sources in a hyperlinked environment. J ACM 46(5):604–632

Levenshtein VI (1966) Binary codes capable of correcting deletions, insertions, and reversals. Technical Report 8, Soviet Physics Doklady

Liu B (2007) Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications), 1st edn. Springer

Maes S, Tuyls K, Vanschoenwinkel B, Manderick B (1993) Credit card fraud detection using bayesian and neural networks. In: Maciunas RJ (ed) Interactive image-guided neurosurgery. American Association Neurological Surgeons, pp 261–270

Newman MEJ (2001) Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality. Phys Rev E 64(1):016132

Newman MEJ (2003) The structure and function of complex networks. SIAM Rev 45(2):167–256

Ni Q, Lobo J, Calo S, Rohatgi P, Bertino E (2009) Automating role-based provisioning by learning from examples. In: The Proceedings of the 14th ACM symposium on Access control models and technologies (SACMAT 2009)

Patriquin A (2007) Connecting the social graph: member overlap at opensocial and facebook, http://blog.compete.com/2007/11/12/connecting-the-social-graph-member-overlap-at-opensocial-and-facebook/

Perols J, Chari K, Agrawal M (2009) Information market-based decision fusion. Manage Sci 55(5):827–842

Saravanan M, Prasad G, Karishma S, Suganthi D (2011) Analyzing and labeling telecom communities using structural properties. Soc Netw Anal Min 1:271–286

Schwartz B (2010) How to connect twitter to facebook status updates. http://www.ehow.com/how_4668396_connect-twitter-facebook-status-updates.html

Scott J (2011) Social network analysis: developments, advances, and prospects. Soc Netw Anal Min 1:21–26

Shehab M, Bertino E, Ghafoor A (2005) Secure collaboration in mediator-free environments. In: Proceedings of the 12th ACM conference on computer and communications security, CCS '05. ACM Press, New York, pp 58–67

Shehab M, Cheek G, Touati H, Squicciarini AC, Cheng PC (2010) Learning based access control in online social networks. In: Proceedings of the 19th international conference on World wide web, WWW '10. ACM, New York

Short JF, Hughes LA (2006) Studying youth gangs. AltaMira Press, New York

Sinclair C, Pierce L, Matzner S (1999) An application of machine learning to network intrusion detection. Computer Secur Appl Conf Ann 0:371

The JUNG Framework Development Team: Java Universal Network/Graph Framework (2009) http://jung.sourceforge.net/

The University of Waikato: WEKA Machine Learning Project (2009) http://www.cs.waikato.ac.nz/ml/index.html

Wang G, Chen H, Atabakhsh H (2004) Automatically detecting deceptive criminal identities. Commun ACM 47(3):70–76

Wang GA, Chen H, Xu JJ, Atabakhsh H (2006) Automatically detecting criminal identity deception: an adaptive detection algorithm. IEEE Trans Syst Man Cybern Part A 36(5):988–999

Wasserman S, Faust K (1994) Social network analysis: methods and applications, illustrated edn. Cambridge University Press, Cambridge

Witten IH, Frank E (2005) Data Mining: practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, San Francisco. ISBN: 0120884070

Xu J, Wang G.A, Li J, Chau M (2007) Complex problem solving: identity matching based on social contextual information. J Assoc Inform Syst 8(10):525–545 (Article 2)