PRIVACY POLICY GENERATION FOR IMAGES IN ONLINE SOCIAL NETWORKS

by

Abdurrahman Can Kurtan

B.S., Computer Engineering, Boğaziçi University, 2016

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2018

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitudes to my thesis supervisor, Prof. Pınar Yolum Birbil, for her valuable guidance, endless support and encouragement. I have learned so much during the two years I have worked with her. Hopefully, I will continue learning in the near future.

I would like to thank my thesis committee members Prof. Levent Akın and Assoc. Prof. Özlem Durmaz İncel for accepting to be in my thesis committee and their comments to improve my research.

I have met with great people in the Computer Engineering Department. I would like to thank Mert İmre, Mert Yaşin, Mert Tiftikçi, Serkan Buğur and Berkant Kepez for their valuable friendship, support and coffee breaks. I would also like to thank Nadin Kökciyan for her contributions and friendship.

I want to thank my dear friends Berke Ersoy, Şenol Özkan, Görkem Sağlam, Büşra Talayman and Ecenur Tuç for being a part of my life. I am also profoundly thankful to Edanur Turna for her support and encouragement. She has always stood by me and motivated me to work.

Finally, I would like to express my gratitude to my dear parents and my sisters for their love and endless support. They have been next to me whenever I needed. I'm deeply grateful to my sisters for trusting me fully and believing in my success. I consider myself lucky to have such a wonderful family.

ABSTRACT

PRIVACY POLICY GENERATION FOR IMAGES IN ONLINE SOCIAL NETWORKS

Image sharing is a service offered by many online social networks. In order to preserve privacy of images, users need to think through and set the privacy settings for each image that they upload. This is difficult for two main reasons: First, research shows that many times users do not know their own privacy preferences, but only become aware of them over time. Second, even when users know their privacy preferences, specifying these policies is cumbersome and requires too much effort, interfering with the quick sharing behavior expected on an social network. Accordingly, this thesis proposes an agent-based approach, PELTE, that predicts the privacy setting of images using their content tags. Each user agent makes use of the privacy settings that its user have set for previous images to predict the privacy setting for a new uploaded one automatically. When in doubt, the agent analyzes the sharing behavior of other trusted agents to make a recommendation to its user about what is private. Contrary to existing approaches that assume a centralized online social network where privacy is set by accessing all the available images, PELTE is distributed and thus each agent can only view the privacy settings of the images that it has shared or those that have been shared with it. Our simulations on a real-life dataset show that PELTE can accurately predict privacy settings even when a user is new in a online social network, she has shared a few images with others, the images have only a few tags or the user's friends have varying privacy preferences.

ÖZET

ÇEVRİMİÇİ SOSYAL AĞLARDAKİ GÖRÜNTÜLER İÇİN MAHREMİYET POLİÇESİ ÜRETİMİ

Görüntü paylaşımı, birçok çevrimiçi sosyal ağ tarafından sunulan bir hizmettir. Görüntülerin mahremiyetini korumak için kullanıcıların, yükledikleri her görüntünün mahremiyet ayarlarını düşünmesi ve ona göre düzenlemesi gerekir. Ancak bu, iki ana nedenden ötürü zordur: Birincisi, araştırmalar, kullanıcıların çoğu zaman kendi mahremiyet tercihlerini bilmediklerini, ancak zaman içinde bunların farkına vardıklarını göstermektedir. İkincisi, kullanıcılar mahremiyet tercihlerini bildiklerinde bile, bu politikaları belirlemek zahmetlidir ve bir sosyal ağda beklenen hızlı paylaşım davranışına müdahale etmek çok fazla çaba gerektirir. Bu tezde görüntülerin içerik etiketlerini kullanarak görüntüler için mahremiyet ayarı öngören etmen tabanlı bir yaklaşım olan PELTE'yi öneriyoruz. Her kullanıcı etmeni, kullanıcısı tarafından yeni yüklenen görüntünün mahremiyet ayarını otomatik olarak tahmin etmek için önceki görüntülerin mahremiyet ayarlarını kullanır. Etmenler şüpheye düştükleri zaman, mahrem olanla ilgili kullanıcıya bir öneride bulunmak için diğer güvenilir etmenlerin paylaşım davranışlarını analiz ederler. Çevrimiçi sosyal ağdaki mevcut tüm görüntülere erişerek mahremiyet ayarı önerisinde bulunan mevcut merkezi yaklaşımların aksine, PELTE etmen temellidir ve böylece her bir kullanıcı etmeni yalnızca kullanıcısının paylaştığı veya kullanıcısıyla paylaşılan görüntülerin ayarlarına erişebilir. Gerçek veri kümeleri kullanarak yaptığımız benzetimler, PELTE'nin bir kullanıcı sosyal ağda yeni olduğu, diğer kullanıcılarla sadece birkaç görüntü paylaştığı, kullanıcının sosyal ağda tanıdıklarının değiştiği ya da görüntülerin sadece birkaç etiketinin olduğu zamanlarda bile mahremiyet ayarlarını doğru bir şekilde tahmin edebildiğini göstermektedir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF TABLES . . . ix
LIST OF ACRONYMS/ABBREVIATIONS . . . x
1. INTRODUCTION . . . 1
1.1. Limited Data . . . 2
1.2. Privacy Variance . . . 3
1.3. Cold Start Problem . . . 3
1.4. Dynamism . . . 4
1.5. Performance . . . 4
2. INFERRING PRIVACY FROM TAGS . . . 6
3. ESTIMATION OF PRIVACY SETTING . . . 15
3.1. Estimation from Internal Data . . . 20
3.2. Estimation from External Data . . . 22
4. EVALUATION . . . 24
4.1. Simulation Environment . . . 24
4.1.1. Private Recall . . . 25
4.1.2. Public Recall . . . 26
4.1.3. Accuracy . . . 26
4.2. Performance Experiments . . . 26
4.2.1. Internal Estimation . . . 27
4.2.2. External Estimation . . . 28
4.2.3. Performance under Privacy Variance . . . 31
4.2.4. Tag Analysis . . . 34
4.2.5. Overcoming Cold-Start Problem . . . 36
4.3. ReBAC Experiments . . . 38
4.3.1. Dataset Creation . . . 38

4.3.2. Performance 41
5. DISCUSSION 47
REFERENCES 57

LIST OF FIGURES

Figure 3.1. Tag Table Update Algorithm 19

Figure 3.2. Privacy Setting Estimation 21

Figure 3.3. External Estimation 23

Figure 4.1. Accuracy of the internal estimation 27

Figure 4.2. The results of the full system against various threshold values . . . 29

Figure 4.3. Results of the full system for 0.01 threshold value 30

Figure 4.4. Accuracy of an agent with 18 friends 32

Figure 4.5. Average accuracy of all agents 33

Figure 4.6. Accuracy against different number of tags per image 34

Figure 4.7. Co-occurrence matrix of most frequent 25 tags 39

Figure 4.8. Algorithm of Group Creation 40

LIST OF TABLES

Table 3.1. An example tag table with three types of relationship 16

Table 4.1. Frequencies of Top Five Tags 35

Table 4.2. Confusion matrix for a new comer 37

Table 4.3. Tag groups of the most frequent 100 tags 42

Table 4.4. Tag groups of the most frequent 100 tags without people and no person 43

Table 4.5. Performance of PELTE on ReBAC experiments 46

LIST OF ACRONYMS/ABBREVIATIONS

OSN	Online Social Network
ReBAC	Relationship-Based Access Control
SNS	Social Network Service

1. INTRODUCTION

Online Social Networks (OSNs) are web-based services where individuals construct a profile to connect, interact and share information with other users within the system [1]. Privacy is one of the friction points that emerges when communications get mediated in OSNs [2]. It has been a concern for humans before the explosive growth of the OSNs [3]. In the second part of the 20th century, Alan Westin [4] defined privacy as "the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information them is communicated." Since OSN services increase the frequency and change the nature of interaction and communication, privacy has became an important concept of OSNs.

OSNs provide personal spaces to people to share their contents, such as images, news items, and so on. Most of the time, users prefer to share their contents with the audience that they see fit. To facilitate the sharing process, users are allowed to define the privacy settings of their content. The current OSNs provide different privacy mechanisms to let users specify their own privacy preferences. Some of them, such as Facebook, let users to specify a set of privacy rules in general. Then it enforces the same privacy rules to specify privacy settings of all images shared by the user. In addition to that, changing privacy settings of an image is also possible if it is desired. Enforcing a set of rules to all images is an easy way to perform a privacy mechanism. However, specifying a general privacy setting for all the images may cause both undesirable accesses to some of those and an unnecessary strictness for some others. Therefore, users can prefer to specify different privacy settings for different sets of images instead of a general one.

Since OSN users have different type of relationships with their connections in real life, users may want specify more customized privacy settings based on relationship types rather than binary privacy settings, which are either deny or permit for everyone. Relationship-Based Access Control (ReBAC) model enables users to specify privacy settings based on interpersonal relationships [5, 6]. A user can categorize her

connections and specify fine-grained privacy settings for different relationship types. However, manually managing these privacy settings is difficult.

Various studies show that OSN users have even difficulties in understanding, let alone, setting the privacy settings of OSNs [7, 8]. Asking a user to manually set a privacy setting every time she is sharing an image will be time consuming and error prone. That is, the user will have to consider all the privacy implications of the image for various audience groups and then set the policy. Second, it is possible that the user does not know which privacy settings are appropriate for a content. This is especially true for new users in the system [9]. Even though OSNs have received much attention in the past years, statistics show that they still continue growing. For instance, the number of active Facebook users had surpassed 1 billion in the third quarter of 2012. As of the fourth quarter of 2017, Facebook had 2.2 billion monthly active users [10]. There are always newcomer engagement and participation in OSNs. Therefore, we aim to help both experienced and new users of OSNs in privacy management. Automated or semi-automated approaches can help users to manage their privacy by predicting the privacy setting for a new image that the user wants to share.

Content of an image can affect a user's privacy concerns about the image. Content based features of an image can be represented by content tags. When images are the subjects, automated systems can use tags to define access-control policies [11]. Content tags of an image can be generated automatically by tools. Recently, different approaches for privacy settings prediction of images have been proposed [12–16]. These approaches partially ignore some important properties that should exist in an automated system that helps users set privacy settings.

1.1. Limited Data

A large body in the literature aim to build centralized classifiers that learn from large image datasets [12–14, 16]. Using all available images in a system—independent of who has shared the image—can provide a plethora of images and thus helps with the classification. However, it is unrealistic to assume that a single entity can access

the images and the privacy settings of all users in the system. For example, a user can see her images and the images shared with her on Facebook but cannot see many of the other images that were not shared with her. Hence, it is necessary to be able to provide estimations without requiring access to all images in the system. Further, it is best if the estimations are based on the image content only, rather than other user details, as such information might not be available in many online social networks.

1.2. Privacy Variance

Another drawback of centralized approaches is that they assume that all users share the same understanding of privacy; thus, all user images are classified in a single shot. However, since privacy is by nature subjective [17], a general classifier that learns from all users' images cannot accurately predict a user's privacy preferences on an image. Therefore, more recent works focus on personalized models [15,18], where the privacy settings for images are estimated based on data from a single user. However, personalized models usually suffer from a cold start problem, initially users do not have enough data to make reliable estimations.

1.3. Cold Start Problem

Ideally, the system should work well even when a user has not shared many images before. For such cases, the system should provide mechanisms for users to make use of the other agents' settings. However, as mentioned above, privacy is subjective. To be able to choose from agents that are trusted for their understanding of privacy is necessary. In multiagent systems, trust is generally defined as a measure of whether an agent acts as expected or not. In the context of image sharing, trust can be used as a measure to interpret how similar a user's privacy preferences with a friend. In order to address privacy variance and cold start problem, Zhong *et al.* [15] propose the first personalized model that is based on user profile data. However, that approach has a highly complex preprocessing step to group similar users via their profile attributes and thus cannot adapt to the users' changing privacy preferences or new users' preferences on images as the environment evolves. Therefore, this approach ignores another

important property of OSNs: dynamism.

1.4. Dynamism

Online social networks enable agents to form new friendships as well as remove old ones. Even when the friendships persist, their strength may vary. Moreover, as with the change in the environment, a user's privacy understanding may change. The system should be able to adapt to these changes immediately. Existing systems [12,16] that predict privacy settings generally ignore this dynamism because they are based on an initial preprocessing phase that defines limited number of private objects for classification process. However, the system should automatically be updated with new information. For example, when a new user enters the system or an existing user shares images with different contents, with each image that they share, the system should be aware of the changes and able to infer their privacy preferences better.

1.5. Performance

An error made in preserving privacy can lead to catastrophic consequences. Hence, it is important to be able to set the privacy policy of an image correctly. Ideally, the system should perform well even when a user has shared few images or little information is known about the images. Because computation power and the knowledge of an agent is limited in a distributed system, it is unrealistic to assume that an agent can have complex learning models.

Accordingly, this thesis proposes an agent-based approach, PELTE, where each user in the system is represented with an agent that helps its user set the privacy of her images. To do so, each image is automatically tagged (by a tool). Each agent uses the tags associated with already-shared images of its user to estimate the privacy policy for new images. However, when a user is new, she might not have shared sufficient number of images, leading to a cold-start problem. When that is the case, the agent uses the images shared by trusted agents to infer privacy policy. Using only the tags of images, PELTE is able to predict privacy policy of images accurately, even when the

number of tags or the number of shared images is low. Furthermore, PELTE is able to accommodate the fact that friends of a user might have varying privacy preferences.

Chapter 2 describes how we infer privacy from tags. Chapter 3 explains our method to estimate privacy setting of images and how we implement the method. We evaluate PELTE in various experiments and present the results in Chapter 4. Finally, in Chapter 5, we discuss our approach in relation to the existing works in the literature and explain the future directions.

2. INFERRING PRIVACY FROM TAGS

We design a system PELTE for OSNs, where each user is represented by an agent that assists the user in managing her privacy while sharing images. Users can share images themselves and can view images that are shared with them over time. When a user is deciding to share an image, she needs to decide with whom the image should be shared. Agents of PELTE aim to support their users in privacy management. Therefore, we can improve user experience while using OSNs.

Definition 2.1. *An agent is a software that represents a user and helps the user preserve privacy when sharing posts. A is the set of agents in the system.*

Agents can have various types of connection between each other. Most of the OSNs that we use today support only one type of relationship. For example, Facebook enables two users connect with each other only as friends. Whereas an agent has one relationship type with a group of agents, it might have another type of relationship with another group of agents. To increase the granularity of privacy settings, different types of relationship, such as Friend, Colleague, Family, and so on, are supported by PELTE. Definition 2.2 captures this.

Definition 2.2. *r^t_{ab} denotes a unidirectional relationship of type t from agent a to b. a_A denotes the set of agents that agent a has relationships with. There are two functions related with relationships.* $\mathsf{subr}(a,t)$ *returns a subset of a_A according to given relationship type t and* $\mathsf{getr}(a,b)$ *returns the set of relationship types from agent a to b.*

Users can share posts on their OSN accounts. These posts can have various types of content, such as text, sound, location, image, video, and so on. For instance, Facebook users can send posts on their timelines to share any type of content with the users in their network. In this thesis, we deal only with images. The aim of the agents is to help users preserve their privacy when sharing images. We assume that each image i contains a set of tags i_T that reflects the content of an image. A tag is

a keyword such as "woman" or "beach" that either identifies an object in the image or reflects a context. These tags might have been produced by the users as well as an automated tool.

Definition 2.3. *A post is a tuple,* $p_{a,i} = \langle c_j^t, ps_k \rangle$ *where its content is* c_j^t *and* $t \in C^{type}$. C^{type} *is the set of content types. It is shared by agent* a *and* ps_k *is the privacy setting of the content.* a_P *denotes the set of posts that are shared by agent* a *and* a_S *denotes the set of posts that are shared with agent* a.

In principle, a privacy decision that sets with whom the post should be shared can contain various audience groups, sets of users, etc., but here we consider the privacy settings in the type of Relationship Based Access Control (ReBAC) that is an approach considering interpersonal relationships between users to regulate accesses [5]. Social networks are suitable to ReBAC since they can be represented as graphs whose vertices are individuals and edges are relationships between individuals. In the graph representation, having different edge types makes system poly-relational. In this type of social networks, the privacy setting related to a post can consist of actions, which are either deny or permit, for each relationship type.

Definition 2.4. *A privacy setting of a post is a tuple* $ps = \langle (r_1, s_1),\ (r_2, s_2),\ ...,\ (r_n, s_n) \rangle$ *where the relationship type* $r_i \in R$ *and sharing action* $s_i \in \{0, 1\}$ *that is either 1 for permit or 0 for deny.* $\mathsf{getAction}(ps, r)$ *is a function that returns the sharing action of the privacy setting* ps *for relationship type* r.

PELTE aims to estimate the privacy setting of an image using its content as opposed to its metadata or other personal information of the user. Content based features of an image can be represented by its tags. When images are the subjects, automated systems can use their tags to define privacy policies [11]. Following this idea, here we use the set of tags i_T an image has. An agent can access the tags of the images its owner has shared or has been shared with it. Each agent uses the tags to decipher what its user finds private.

Users can upload images of various contexts to their OSN accounts. An image might be considered as appropriate to share with everyone in the user's network. In that case, the user permits every relation type while specifying the privacy setting of the image. However, if the image has a context that relates to a specific group of audience, then the image owner may prefer sharing the image only with that group of users. For example, an image of a business meeting probably might be considered as only related to users that are colleagues. Therefore, the user can choose a privacy setting that permits only the users having the relationship type of Colleague. Moreover, some images might involve sensitive information about the user. In that case, user can prefer to share the image only with a small group of audience that is close enough to see the image. For example, a teenager might prefer to share his party pictures with his close friends at the party, but might not want them to be seen by his family or teachers.

Users specify their privacy settings according to what they consider as private and what they do not. If a user has consistent decisions about what to share with a type of relationship, we can observe patterns in the tags of images that the user permits the given relationship type. Conversely, if the user does not share images of particular contexts, possible tags relating to these contexts cannot be found in the user's images. However, if a tag is in the tag list of both public and private images of the user, we cannot be sure about the relation between the tag and the privacy decision even if there is any. Consequently, the privacy setting of a newly uploaded image can be estimated with the consideration of user's previous tags. Moreover, we can be more sure about the privacy prediction if the user's previous decisions about the images of similar contexts are consistent and can be seen multiple times. We can reveal these patterns between privacy understanding of users and their standpoint against relationship types. Then, we can use the patterns to estimate privacy settings of new uploaded images. In order to do that, we need to model the patterns computationally. If the user has previously shared too many images before, an obvious choice would be to use machine learning techniques. However, since our agents rely on local set of images only and especially aims to help users who are new in the system, the available data set is expected to be small. For these reasons, we resort to methods that are inspired from information

retrieval, where we measure the influence of tags for images. We propose two metrics for tags that are relevant for this purpose: *support* and *effect*.

Support value of a tag shows the number of images that have the tag. If there are many images with the same tag, we can know the privacy preference on a tag more strongly. That is, higher *support value* reveals more precise information about the user's privacy preferences on the content. Equation 2.1 describes how the support of a tag t is determined for a user a where a_I is the set of images that a has shared.

$$S_{a,t} = \sum_{i \in a_I} \mathbb{I}(t \in i_T) \tag{2.1}$$

Effect value of a tag denotes the number of shared images that have the tag in their tag list i_T. Since we use ReBAC in privacy settings, each relationship type has its own effect value. Equation 2.2 shows the calculation of an effect value of a tag. Normalization of the *effect value* of a relationship type with its *support value* yields the ratio of images that permits the relationship type to all images with the same tag. The result is between 0 and 1. If the value is smaller, images are mostly not shared with the users with the given type of relationship. Conversely, the value is close to 1 if the given relationship type is permitted for most of the images.

$$E_{a,t,r} = \sum_{i \in a_I} \mathbb{I}(t \in i_T) * \mathsf{getAction}(i_{ps}, r) \tag{2.2}$$

The effect value denotes how strongly the tag is considered to be private against a relationship type. The values that are around 0.5 show that while many images having the tag are shared with the user of the relationship type, many others are not shared.

Therefore, we conclude that the user's privacy preference on the tag is inconsistent and the tag is not too informative for future estimations. Thus, tags having high and low effect values give more precise information about the user's privacy preferences.

Above mentioned properties clearly indicate that tags with high *support value* and *effect value* that is either close to maximum or minimum value are valuable to estimate user's privacy preference for an image. In two cases, user's privacy preference for a tag may not be clear. In the first one, the tag may not be in the overall set of tags if the user has not shared an image with the tag yet. In the other case, the user may have already shared many images with the tag but the effect value of the tag can be close to the average since the user's previous privacy decisions are not consistent with respect to the tag.

Confidence value is a metric to infer privacy settings of an image i from its tags. It is calculated for each relation type r, separately. i_T is the set of tags of the image, a_T is the overall set of tags that are collected from the images of agent a.

$$Conf(a,i,r) = \frac{\sum_{t \in i_T} E_{a,t,r} + |i_T \setminus a_T| \dfrac{\sum_{t \in a_T} E_{a,t,r}}{|a_T|}}{\sum_{t \in i_T} S_{a,t} + |i_T \setminus a_T| \dfrac{\sum_{t \in a_T} S_{a,t}}{|a_T|}} \tag{2.3}$$

Equation 2.3 calculates the effect per support value for the tags that are associated with a given image. In doing so, it first evaluates the total effect of image tags that the agent has seen before (i.e., in the agent's overall set of tags). However, it also takes into account the image tags that the agent has not seen so far by assuming their values to be average *effect values* and *support values*. Taking these tags into account results in that the metric to yield values that signal an uncertain privacy setting. This is a desired outcome because the agent has no previous experience on these tags and thus should be cautious in estimating the privacy setting by using them.

As we mentioned before, having a confidence value that is not around the average value is more valuable to infer privacy. Therefore, the confidence value is meaningful only when it is compared with average effect value per support. It is calculated as follows:

$$Avg(a,r) = \frac{\sum_{t \in a_T} E_{a,t,r}}{\sum_{t \in a_T} S_{a,t}} \tag{2.4}$$

Since it is possible to compare a *confidence value* of tags of an image and *average value* of all tags via Equation 2.3 and Equation 2.4, the system can infer the privacy setting of an image for each relationship type. If the *confidence value* of the image is higher than the *average value* for a given relationship type, the image would be considered more probable to be shared with the given relationship type. However, as mentioned before, confidence values that are close to average could easily be misleading. To signal this to the agent, we use a threshold θ and require that the confidence is at least θ amount different than the average. Equation 2.5 formalizes this intuition. Notice that there will be cases when the estimation will not reliably conclude with either share or do not share action. In this case, PELTE analyzes the sharing behavior of other individuals in the system. However, these are not random individuals from the network but those that the user has social ties with e.g., friends that have extensively shared images of similar contents having the same privacy settings with the user.

$$Est(a,i,r) = \begin{cases} 1 & if\ Conf(a,i,r) > Avg(a,r) + \theta \\ 0 & if\ Conf(a,i,r) < Avg(a,r) - \theta \\ Ext(a,i,r) & otherwise \end{cases} \tag{2.5}$$

The intuition of that $Ext(a, i, r)$ part of the *estimation* function comes from Social Learning Theory [19]. It suggests that people learn by observation in social situations, and that they begin to act like the people they observe. Burke *et al.* [20] study the social learning theory in OSNs to test the extent to which social learning motivates content sharing by new members. They analyze the data coming from 140,292 Facebook profiles, using information from their first three months of Facebook membership. They reveal that new members are closely monitoring and adapting to what their friends are doing. This research demonstrates that social learning is an important influence on new users. New members closely follow the actions of their friends on OSN and adapt their content contributions accordingly.

We are inspired from the Social Learning Learning in the sense that agents mimic their friends if they do not have certain privacy preferences while sharing images. This usually happens if the user is a newcomer. From the perspective of a newcomer, an OSN is a union of previously joined users and the contents that the users have already created. As new users start to share their own images, they build their own privacy preferences over time. This is in line with the *support value* of PELTE. However, users may not have a certain context if they are not newcomers. In that case, we consult the Social Learning Theory again. This time, since users are not completely inexperienced, they may adapt more some of their friend whereas ignoring some others.

In practice, an agent would benefit most from friends that share the same privacy preferences. For example, if two friends always share images with similar tags, this would signal that their privacy preferences are similar. Based on this intuition, in PELTE, each agent analyzes its friends' privacy settings of their shared images to judge how similar they are to each other. Agents with similar privacy preferences are favored when obtaining privacy opinions.

Agents compare their privacy settings with their friends and calculate the similarity between privacy preferences. The resulting value is the trust value of agents towards their friends. In this context, trust is used as a measure to calculate how similar a user's privacy preferences are with a friend. Nevertheless, trust might be

calculated in different ways. In OSN environments, as profiles are attributed to presumably known persons from the real world, they are implicitly valued with the same trust as the assumed owner of the profile [21]. However, real world experience cannot be directly transmitted to a virtual environment as a numerical value. Even though users can manually assign trust values to their friends in a social network, this is not a feasible way to do for an automated system.

In OSNs, as relationships develop, trust becomes important value in building self-confidence. It is possible that over time, the role of trust become even more important in OSNs, where personal information exchange occurs [22]. In PELTE, agents only have limited knowledge about each other and they improve their knowledge bases over time as they share more images with each other. They can accumulate the data coming from shared images. From this point of view, we create a trust metric specific to images.

Equation 2.6 measures the trust value of agent a to its friend agent b based on a set of images, b_I, which b has shared with a. Our trust metric is multidimensional, where each dimension corresponds to trust towards agent for a type of relationship. For each type of relationship, it compares the privacy setting of the image (as set by agent b) and the sharing action agent a would have taken (using Equation 2.5), if agent a was actually sharing the image. The trust in an agent increases when the number of images with same privacy preferences is high. Since a user does not have privacy preferences when she is a newcomer, the user cannot measure the trust towards her friends. Therefore, each user completely trusts friends at the beginning. This corresponds to the observation phase of the the Social Learning Theory and thus the users adapts to their friends. Note that the trust values are unidirectional; that is, agent a's trust in agent b could be different than agent b's trust in a. Moreover, trust towards the same agent might be different for different each types of relationship.

$$trust_{a,b,r} = \frac{\sum_{i \in b_I} \mathbb{I}(Estimation(a, i, r) = \mathsf{getAction}(i_{ps}, r)}{|b_I|} \tag{2.6}$$

Huynh *et al.* [23] develop a model that integrates four different types of trust and reputation in a multiagent system. They use interaction trust, role-based trust, witness reputation, and certified reputation together. Their model combines values coming from different trust mechanisms and produces one result. Variations of their model could be employed in our system to enhance the trust metric. One approach could be a metric that benefits from different sources addition to the similarity of privacy preferences. For example, profile attributes of users or interactions other than image sharing could be beneficial for trust calculation. Another approach could be that using different trust metrics on each dimension of the trust. Multidimensional structure is suitable to use different metrics individually. For instance, a relation type like colleague may have the data of user roles. If this is the case, a metric similar to the role-based trust could be employed to infer expertise of users from their roles. However, we design and use a basic trust metric because we assume that only the data related to images are available on the system. If, in the future more information becomes available, PELTE can support an extended trust metric that takes into account the relation-based trust. We leave this as an interesting future direction.

Finally, PELTE estimates the privacy setting of each image via the functions we explain above. According to system design, the resulting setting can be used by the agent to automatically set the privacy setting of the image or to be suggested to the user as a decision suggestion. In either cases the system helps user preserve her privacy and make the image sharing process easier. Since designers of social network services seek to improve the overall experience by encouraging members to contribute more content [20, 24], PELTE can be considered as a desirable and important support system for social network services.

3. ESTIMATION OF PRIVACY SETTING

In order to estimate privacy setting of an image that will be shared by the user, the support and the effect values of tags should be processed. To make them easily computable by the software, we store the data in *tag table*s. *Tag table* is a modified version of two dimensional array. Each row of the *tag table* corresponds to a tag, where for each tag its name t, its support value $S_{a,t}$ and its effect values $E_{a,t,r}$ are stored in different columns. The table is indexed by tag names. In PELTE, each agent stores its all knowledge about the shared images in this data structure. A different way to store the data would be to record tags of images separately as lists. Because same tags are possibly found in the lists of different images, this would be inefficient in terms of the space complexity. This is, the space complexity of the *tag table* structure is proportional to the number of unique tags, whereas storing the data of all images as list is proportional to both the number of images and the average number of tags an image has. Since the number of unique tags are much less than the total number of tags, tag table is a highly efficient way of storing the data of images.

An example tag table is presented in Table 3.1. The first row of the table shows the column names. There are three types of relationship type, namely Friend, Colleague, and Family, in the environment. Effect values of tags for each relationship are in the columns under the names of relationship types. The given tag table is just a small part of a bigger one and sorted according to the support value, but normally tags do not have to in be that order.

For example, the first row of the table shows that the user has shared 95 images having “people” in their tag list. The user shares 12 of these images with her friends. Her colleagues can access to only 10 of these images. However, she permits her family to access 42 of them. “People” is the most frequent tag of the table. The tags at the bottom rows of the table have support value of 1. These tags can be rarely seen in the

Table 3.1. An example tag table with three types of relationship

Tag name	Support Value	Friend	Colleague	Family
people	95	12	10	42
woman	71	5	0	25
adult	70	6	2	28
portrait	69	6	4	23
one	63	10	7	27
girl	45	3	2	11
fashion	35	6	1	5
indoors	34	3	4	17
child	28	1	2	14
facial expression	19	0	1	6
son	11	1	2	8
brunette	11	0	0	2
nude	10	2	1	3
wall	6	4	4	5
vacation	4	1	1	2
blur	3	1	1	2
hand	2	0	0	1
manicure	1	0	0	1
treatment	1	0	0	1
fingernail	1	0	0	1
bay	1	1	1	1
surf	1	1	1	1
shore	1	1	1	1

user's shared images.

In PELTE, there are two types of tag table: *internal tag table* and *external tag table*. *Internal tag table* stores the data of privacy settings that are collected from images that the user shares herself. Mentioned as the principle of privacy variance in Chapter 1, we aim a personalized model since privacy is by nature subjective. Therefore, *internal tag table* is the key component that makes the system personalized by storing all the data coming from the user herself. *External tag table* stores the data collected from the images that the user's friends' have shared with the user. Differently from *internal tag table*, *external tag table* stores the data coming from the privacy settings of images that are shared with the user. We assume that if a user can view an image, then the user's agent can obtain the privacy settings of the image. For example, images in Facebook have icons indicating the privacy settings of the images to let users know which other users can see their likes and comments on that image. If an image is shared publicly, then we see an icon of world above the image. If there is an icon of people, we understand that the image is shared only with friends. In PELTE, when an agent shares an image, the image is sent to all agents whose relationship type is permitted by the privacy setting of the image. Then, the agents that receive the image stores its tags into their *external tag table*s according the privacy setting relating to the image.

In the design of PELTE, we use maps to store the privacy settings of images. In programming languages, map is an object that maps keys to values. A map cannot contain duplicate keys; each key can map to at most one value. Map has a method called get that returns the value to which the specified key is mapped. According to our privacy setting, which is defined in Definition 2.4, sharing actions are mapped to relationship types. There cannot be more than one sharing action for a relationship type. Therefore, maps meet the properties that we use specify for privacy settings. Relation types are the keys of maps and sharing actions are the values of the keys. Thus, we can simply implement $\mathsf{getAction}(ps, r)$ function in Definition 2.4 by using default get function of maps.

Each agent starts with empty two tag tables, internal and external ones, and an empty list of relationship when it joins the system. Then it adds agents into its list of relationship as it establishes relationships. It collects data from the environment over time. Figure 3.1 shows how the tables are filled as new images are shared. Whenever a user shares a new image, the user's *internal tag table* is updated according to the privacy setting of the image. First, the agent generates tags of the images (line 2). This can be done automatically in various ways, *e.g.*, using the tool, Clarifai.com [25], which provides a service that takes an image and returns up to 20 tags of the image. In the current OSNs, users usually share the images with a set of tags to make them appear in the network. This would be implemented as attaching the tags to the image as a list. However, we generate the tags by using a tool since we do not have the tags initially. After the agent gets the tags of the image, it updates the corresponding rows of the tags in the table. If any of the tags is not already stored in the tag table, it is added to the table (Line 5). Then, *support* and *effect* values of the tags will be updated (Line 7, 10). For each tag in the list, effect values of the relationships that are permitted by the privacy setting are incremented by one. If the image is shared privately against a relationship, the corresponding effect values of the tags remain the same value. In both cases, the support value of each tag is incremented by one.

While the agent is updating its internal tag table, the agents of the users, with whom the image has been shared with, update their *external tag table*s. Now, these agents have the image, which is shared with them, and the privacy setting of it. This time, the same update operations are performed for the *external tag table*. Tag table, a_T, in the algorithm corresponds to the external tag table of the agent instead of internal one. Each of the agents of users who can access the shared image executes the same process.

Trust values coming from Equation 2.6 are used in the update process of external tag table. Trust values are used as the multipliers in the update operation of support (Equation 7) and effect (Equation 10) values. Initial values of trust towards other users is 1. This means that agents completely trust to other users at the beginning. They learn how to act from other agents without judging their actions as proposed by the

Algorithm 1: Tag Table Update(i, a)

Input: i, image

Input: a, agent

1 $a_T \leftarrow \mathsf{getTagTable}(a)$;

2 $i_T \leftarrow \mathsf{getTags}(i)$;

3 **foreach** *tag* t *in* i_T **do**

4 **if** $t \notin a_T$ **then**

5 $a_T = a_T \cup t$; // add tag t to the a_T

6 **end**

7 $S_{a,t} = S_{a,t} + 1$; // update support value

8 **foreach** *relationship type* r *in* R **do**

9 $sa \leftarrow \mathsf{getAction}(i_{PS}, r)$; // get sharing action for r

10 $E_{a,t,r} = E_{a,t,r} + sa$; // update effect with the sharing action

11 **end**

12 **end**

Figure 3.1. Tag Table Update Algorithm

Social Learning Theory. However, trust toward a friend change over time according to similarity between privacy preferences of the user and those of a friend. Trust towards users sharing images with different privacy settings decreases. Thus, the data of images that are shared by users having similar privacy preferences affect more than the data of images that are shared by others. As a result of these update processes, both the *internal tag table* and *external tag table* of the agents in an environment will be dynamically updated. Thus, whenever there are changes in a user's understanding of privacy, *internal tag table* of the user changes accordingly in time. Similarly, if the user's friends' understanding of privacy change over time or she establishes new connections with other users who have different understanding of privacy, then the user's *external tag table* changes over time and adapts to new ideas. However, the change in the tag tables may need time according to the *support* and *effect values* of the data that are already stored in them.

3.1. Estimation from Internal Data

While estimating the privacy settings of a newly updated image, the first option of PELTE is to use the internal tag table of a user. It takes sharing actions against each relationship type based on Equation 2.5. If the confidence value is not around the average value, than the privacy setting can be estimated internally. However, if the confidence value for any of the relationship types is around average, it infers that user's preference on sharing the image with the relationship type is uncertain and leave the action to the external estimation. Thus, even though a user's previous images have not enough data itself, the system can estimate privacy setting of the user's new uploaded image.

The algorithm for the estimation of privacy settings from internal data is presented in Figure 3.2. The system works as follows: It generates tags of an uploaded image by using the tool. Then, it searches the internal tag table for the tags of the image (Line 4). One important point of this search is that it counts the tags that are not found in the table (Line 5, 6). Since these tags are not informative about the privacy of the image, their effect and support are taken as average values (Line 16). To decide whether the image should be shared with a relationship type, it calculates the metric, *confidence value* and compares it with the average effect per support value. If it is higher than the value then decides to share with the given relationship type and adds a permit action to privacy setting (Line 18). Otherwise, it adds sharing action of deny to the privacy setting for the relationship type (Line 20). In the best case scenario, the confidence value is either 0 for the deny action or 1 for the permit action.

If the *confidence value* is around the average value and within the threshold boundaries (Line 21), estimation from internal data cannot return a sharing action for the relationship type. Therefore, it leaves the decision about the relationship type to the estimation from external data.

Algorithm 2: InternalEstimation(a, i)

Input : a, agent; i, uploaded image

Output: PS, estimated privacy setting

$i_T \leftarrow \mathsf{getTags}(i)$;
$avgSupport \leftarrow \mathsf{getAvgSupport}(a_T)$;
$n, support, effect \leftarrow 0$;
foreach *tag* t *in* i_T **do**
    if $t \notin a_T$ **then**
        $n = n + 1$;
    else
        $support = support + S_{a,t}$;
        foreach *relationship type* r *in* R **do**
            $effect[r] = effect[r] + E_{a,t,r}$;
        end
    end
end
foreach *relationship type* r *in* R **do**
    $avgEffect \leftarrow \mathsf{getAvgEffect}(a_T, r)$;
    $confidence = \dfrac{effect[r] + avgEffect * n}{support + avgSupport * n}$;
    if $confidence > \dfrac{avgEffect}{avgSupport} + \theta$ **then** Permit
        $PS.\mathsf{add}(r, 1)$;
    else if $confidence < \dfrac{avgEffect}{avgSupport} - \theta$ **then**Deny
        $PS.\mathsf{add}(r, 0)$;
    else Undecidable
        $PS \leftarrow \mathsf{ExternalEstimation}(i, r)$;
    end
end
return PS;

Figure 3.2. Privacy Setting Estimation

3.2. Estimation from External Data

Systems that are based on making decision based on historical data typically suffer from the cold start problem when the required historical data are not available. In our context, when a user has not shared images with a content, possible tags that can be generated from these images cannot be found in the user's internal tag table. This can occur in two different situations: When a user's internal tag table does not have enough tags because she is new or she has not shared any images with that content. Therefore, estimation from internal data mechanism cannot decide to privacy setting of the image.

To handle the cold part problem in the context of privacy, different approaches such as asking the trusted users for a privacy policy [18] or grouping user based on their profile attributes [15] have been proposed. Asking a user for a privacy policy without analyzing her experience on the content may get a wrong privacy policy. Grouping users may not be possible in an environment where profile data do not exist. Instead of these approaches, we estimate the privacy setting of an image from user's friends' experience on similar images via the data that can be collected by just using shared images.

When the internal estimation cannot decide whether to share the image with a relationship type, the external estimation takes over the job. External estimation uses the algorithm in Figure 3.3, which is similar to the Algorithm in Figure 3.2. However, this time tag table a_T in the algorithm is the external tag table of the user. This algorithm returns a sharing action for the given relationship instead of a complete privacy setting.

Algorithm 3: ExternalEstimation(a, i, r)

Input : i, uploaded image, r relationship type

Output: sa, sharing action for relationship r

$i_T \leftarrow \mathsf{getTags}(i)$;
$avgSupport \leftarrow \mathsf{getAvgSupport}(a_T)$;
$n, support, effect \leftarrow 0$;
foreach *tag* t *in* i_T **do**
    if $t \notin a_T$ **then**
        $n = n + 1$;
    else
        $support = support + S_{a,t}$;
        foreach *relationship type* r *in* R **do**
            $effect[r] = effect[r] + E_{a,t,r}$;
        end
    end
end
$avgEffect \leftarrow \mathsf{getAvgEffect}(a_T, r)$;
$confidence = \dfrac{effect[r] + avgEffect * n}{support + avgSupport * n}$;
if $confidence > \dfrac{avgEffect}{avgSupport}$ **then** Permit
    $sa = 1$;
else
    $sa = 0$;
end
return as;

Figure 3.3. External Estimation

4. EVALUATION

We implement PELTE as a software tool in Java. We evaluate PELTE in different scenarios to show the capabilities of the proposed approach. An important aspect is to show that the proposed approach indeed helps users preserve their privacy under various circumstances. Of particular importance are the properties explained in Chapter 1: limited data (Can PELTE work well if the images have only a few tags?), cold start problem (Can PELTE still predict the right privacy preference, if the user has shared only a few images before?), and privacy variance (Is PELTE prune to cases where some of the agents are reporting contradictory privacy settings?). We evaluate our proposed approach over multiagent simulations to answer these questions.

4.1. Simulation Environment

In order to simulate an online social network, first a social network graph is needed. We construct the network by using the Facebook dataset called ego-Facebook obtained from Standford University Network Analysis Project [26]. The dataset has different sized networks. We use the network that contains 52 nodes and 146 bidirectional, friend relationships among the nodes, where each node might have different number of relations.

The simulation works as follows: it starts with creating an agent for each node and constructs relationships between them. Each agent has two main data structures that correspond to *internal tag table* and *external tag table* defined in Chapter 3. Then the image sharing process starts. During the training phase, privacy settings of images are defined according to the labels defined in the dataset. While distributing images to agents, the simulator shuffles the list of agents and picks one of them randomly. This corresponds to the agent sharing the image itself. After the image coming up next is assigned to that agent, the agent updates its internal tag table. Similarly, its friend agents update their external tag tables. When the training phase ends, privacy settings of new assigned images are estimated from the data in the tag tables

of the agents. This process is the implementation of Algorithm 3.2. Since the image distribution is randomly performed, the number of images each agent has might be different. Moreover, each run of the simulation distributes images to agents in different orders. Therefore, an agent will have a different set of images in separate runs. To reduce the effect of randomness on the results, we run each experiment 20 times and we present average of the calculated values as final results.

The images used in the simulation environment are obtained from PicAlert [27]. It is one of the widely used datasets of image privacy studies. This dataset has 37510 Flickr images and privacy labels, which are collaboratively created by human evaluators via impersonation method. The possible privacy labels are private, public and undecidable. We remove undecidable labels from the dataset all together. The dataset only stores the ids of the images and some of those are not available on Flickr any more. After we remove them from the dataset, we eventually have 29866 different images. An image might have different labels by different evaluators since they are labeled collaboratively. Not to cause false predictions because of the contradictions, we remove images that have opposite labels. Then, we pick all images that have been labeled as private by all the evaluators, totally 3500 private images from the dataset. The total number of images that have been labeled as public by all the evaluators is much more than private ones. Therefore, we randomly sample 3500 public images to keep the size of public and private image sets equal. Each image has 20 different tags, which are generated by the general model of Clarifai [25]. These tags correspond to concepts, objects, scenes, and more.

The performance of the proposed model is evaluated via following success metrics:

4.1.1. Private Recall

As we mentioned before, because people are more concerned with false allows than false denies [11], preserving users' privacy is the main concern of PELTE. Therefore, we calculate how many of the private images are predicted as private and present the result as private recall of an evaluation. Ensuring that the private images are identified

as private is the main task of PELTE.

4.1.2. Public Recall

It is obvious that predicting all images as private can preserve privacy without any mistake. However, users share their personal information on OSNs because they intend to share or transmit information to their friends [28]. But their willingness do share their personal data depends on the sensitivity of the data [29]. Therefore, PELTE is expected to be able to differentiate images that can be shared publicly according to user's privacy preferences and the properties of the image. Thus, it enables users to reach other users as much as possible via their shared images without enforcing unnecessary strictness.

4.1.3. Accuracy

Private and public recalls show success of the system from different aspect. But the overall success of PELTE mainly depends on to be able to set the privacy preferences of each image correctly. Accuracy value represents the ratio of all images that are predicted correctly to the total number of images.

4.2. Performance Experiments

Although our proposed approach aims to work on OSNs where ReBAC is possible, the image and network graph datasets we have just correspond to one relationship type. To clarify, network graph data do not have relationship type in it and the image dataset has just one label for each image. Therefore, our datasets limit the evaluations with one relationship type. We evaluate the performance of the model step by step for each feature it has.

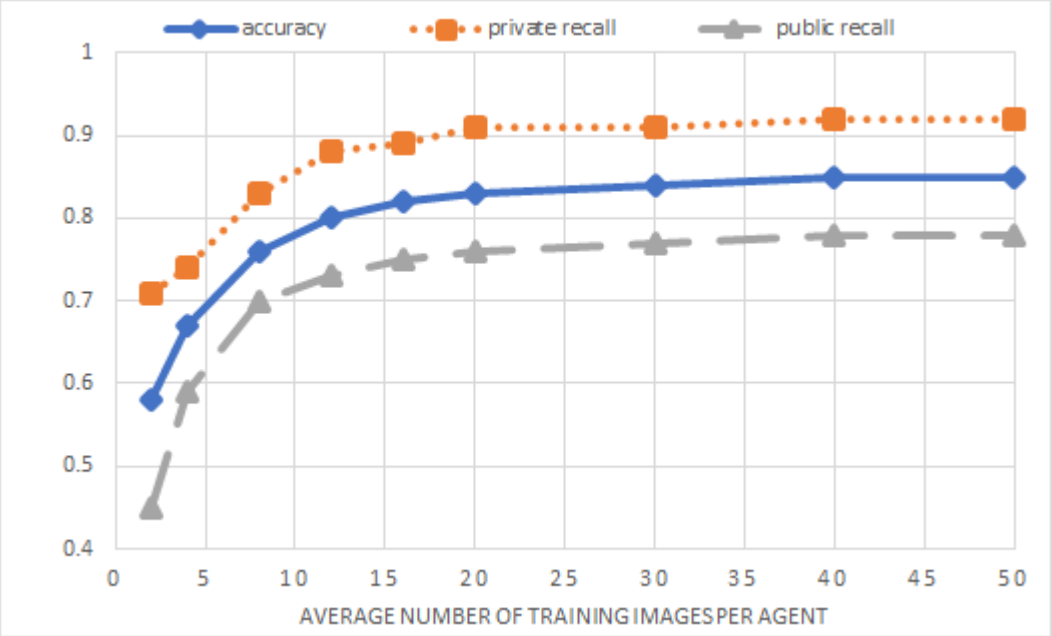

Figure 4.1. Accuracy of the internal estimation

4.2.1. Internal Estimation

First, we evaluate the privacy setting estimation of the system when only internal data are available. This part mainly focuses on the effect of the number of training images on the accuracy and the private and public recall values. In each experiment setup, we evaluate the internal estimation with given number of training images 20 times. Then we change the size of the training data run the simulation 20 times again. We do the same operation for various number of training images and collect the results from each of them. In Figure 4.1, we present the average of the results obtained from each experiment setup. The x axis is the average number of training images per agent in each experiment setup. We then plot accuracy, private recall, and public recall. The corresponding results are for 1000 test images. The system starts with an accuracy value around 0.6 when there are 100 training images throughout the system and each agent has approximately two images. Providing more training data to the agents make the system more accurate, as expected. However, the total number of images that is required to reach a 0.85 accuracy throughout the system is only 2000, such that each user agent has approximately 40 images. These results show that the PELTE successfully estimates the privacy of images even it uses only the users' internal data.

Note that the private recall attains a result that is higher than the accuracy of system and around 0.9. In other words, the system estimates privacy setting of private images more accurately than public images.

Another important point of the results in Figure 4.1 is that PELTE starts from low accuracy and recall results when there are few number of training images and it has an increasing success until it reaches the best accuracy. The increasing success part of the results correspond to where the cold start problem of the system occurs. Because there are not enough number of images to learn users' privacy preferences, it cannot estimate them accurately and makes more mistakes than it does in its best results. As we already mentioned in Section 3.2, we aim to improve these results by using privacy settings of users' friends' images that are shared with them.

4.2.2. External Estimation

In some cases, a user's privacy preference on an image may not be estimated from the user's internal data. Therefore, the confidence value can be close to the average value. These cases are eliminated by using a threshold θ, as shown in Algorithm 3.2 (line 21). Images within the threshold boundaries are directed to the external estimation mechanism. This situation mostly occurs when there are few images in the internal tag table of a user agent. In the experiments of Figure 4.1, because the threshold value is equal to 0, privacy settings of all images are estimated internally. This is, the system works as there are not available external data and all images have to be labeled by the internal estimation. Therefore, the images that have confidence value close to the average value are labeled as either public or private even the estimation is not strong enough. Internally estimated privacy settings of these images are more likely to be incorrect. Hence, the external estimation is expected to improve the results in the labeling of these images.

We analyze the effect of the external estimation on accuracy and recall values by changing the threshold θ in 2.5. The results of the experiments with different values of the threshold θ can be seen in Figure 4.2. The x axis of the figure is threshold θ

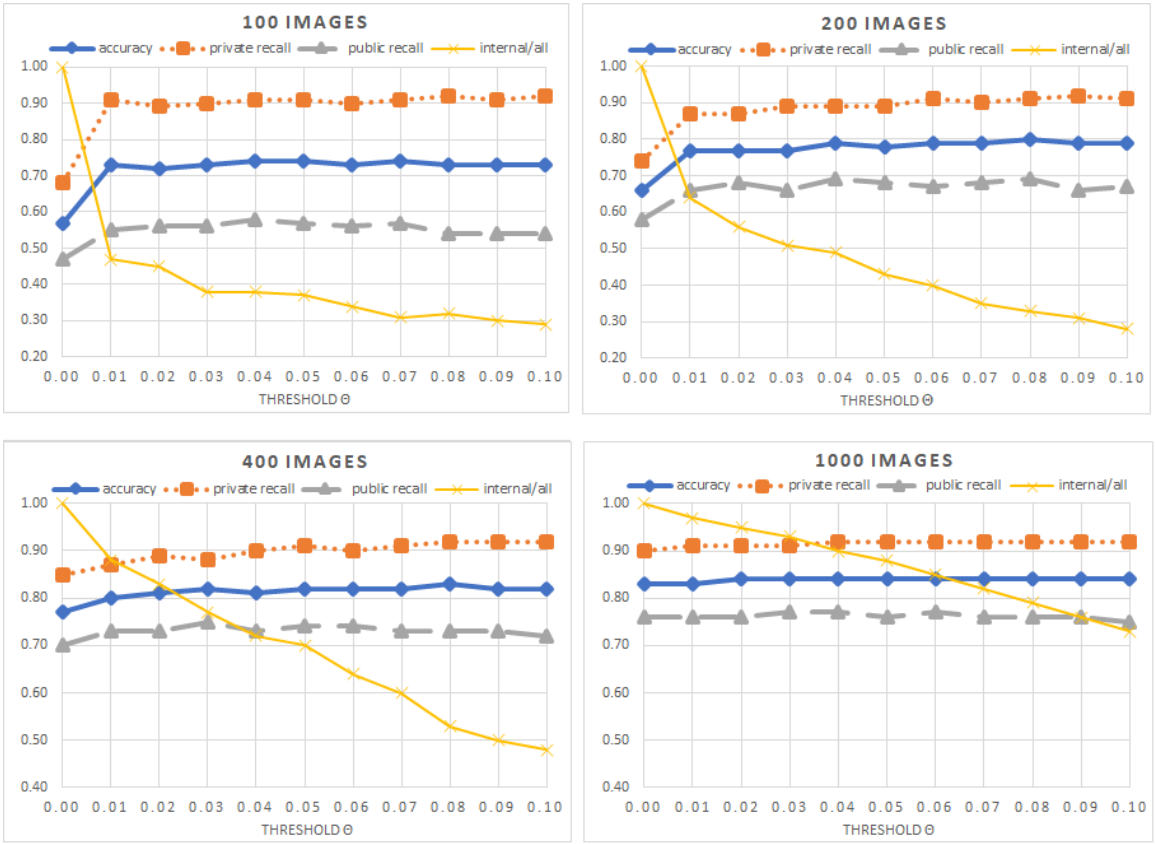

Figure 4.2. The results of the full system against various threshold values

value applied in the experiments. We plot the accuracy and recall results of the overall system when there are 100, 200, 400, and 1000 training images throughout the system. Also, there is *internal/all* line, which shows the ratio of images labeled by internal estimation to all images labeled by the system. It helps us to understand how many images are labeled by the external estimation. Since the system cannot estimate a privacy setting at all when the tag tables are empty, we start with 100 images, which correspond to two images per agent. We observe the results against increasing amount of threshold value. It starts from 0 and increases up to 0.10. Having a threshold value of 0 means it estimates privacy settings of all images internally. 0.01 threshold value is where the external estimation becomes active. The accuracy and the recall values increases when it becomes active. But the results are expected to be dependent how much data are stored in the tag tables. Therefore, we observe it under different number of training images to be sure about the effect of the threshold value. Even though the exact results change with the number of training images, it is obvious that the external

estimation has a positive effect on the accuracy and recall values in all charts. But the main result obtained from these charts is that increasing the threshold value does not always make the system much more successful. Higher threshold value causes more images to have externally estimated privacy setting instead of internally estimated one. This is valid for all experiments and obviously can be seen from the plot of *internal/all* in the charts. Moreover, the model needs external estimation less with the increasing number of images shared by agents themselves.

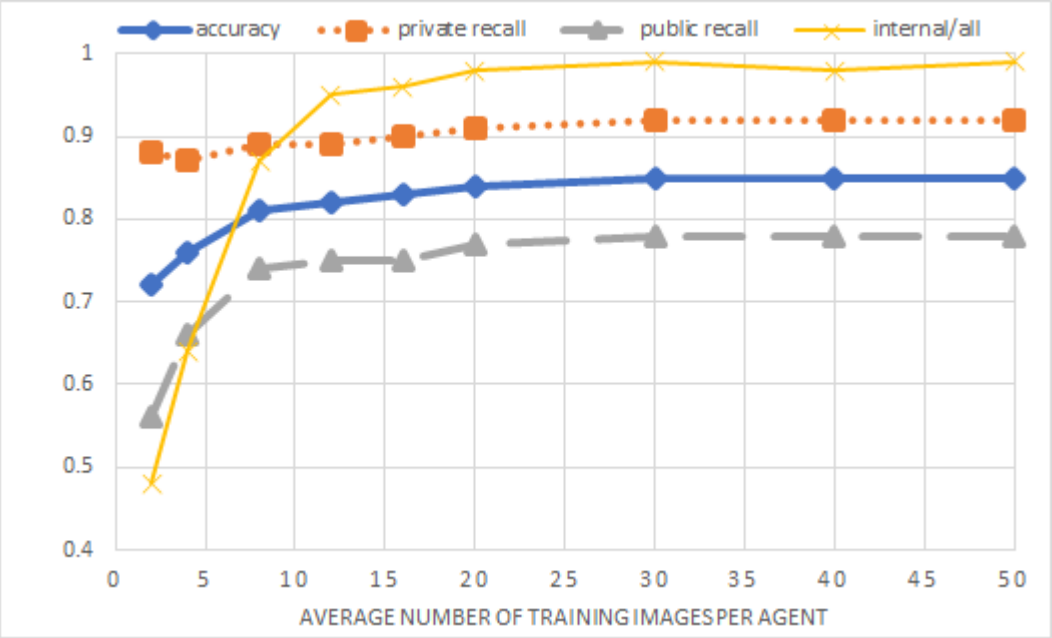

Figure 4.3. Results of the full system for 0.01 threshold value

Now, we know that using small threshold value is enough to increase success of the system via external estimation. Accuracy, private recall, and public recall results of the system having 0.01 threshold θ value against different number of training images are given in Figure 4.3. The x axis is the average number of images each agent has. We plot the accuracy, private recall, public recall and the ratio of images labeled by internal estimation to the all images labeled by PELTE. We see that both accuracy and recall values becomes better with the increase in the number of training images per agent. Moreover, the increase in the *internal/all* values shows that PELTE estimates the privacy settings of more images when agents have more data in their internal tag tables.

We can see the positive effect of the external estimation to the results more clearly by comparing Figure 4.1 and Figure 4.3. When each agent has only two images, the accuracy of the internal estimation is less than 60%. However, when the system enables the external estimation, the accuracy of the system becomes 70%. The system reaches its maximum success earlier than the system that benefits from only the internal estimation. Even when there are few number of training images, accuracy and recall values are comparable to the best results that the full system achieves. Therefore, we can conclude that the privacy estimation from external data improves the results of the system when it suffers from the cold start problem. If the system only used internal data, it would have required much more data to yield the result that is obtained with external estimation.

4.2.3. Performance under Privacy Variance

In the previous scenarios, all agents are assumed to be sharing the same privacy understanding. However, privacy is inherently subjective. For example, wedding pictures could be shared publicly by many while it might be considered private by others. That is, an agent might not prefer to share an image even if many of its friends on the network are sharing similar images. Accordingly, by making decisions based on what others have shared with the agent might give misleading results. Similarly, some friends may not properly know how to share images on OSN. Hence, the agent should only make decisions based on the agents that it has similar privacy understanding. This is represented as the trust metric of PELTE, which is defined in Equation 2.6. In order to show, how well agents make decisions based on trust in PELTE, we construct a scenario, where there are agents with contrasting privacy understandings. These agents share images with opposite privacy settings, *i.e.*, sharing an image as public when the system infers the image is private or visa versa.

We evaluate this scenario from two different perspectives: micro view and macro view. In micro view evaluation, we construct the network as we do in the previous scenarios. We pick an agent and view the network from its perspective. The particular agent chosen has 18 friends in the network. We vary the total number of contrasting

agents among these 18 to see the effect of the trust on privacy estimation from external data only.

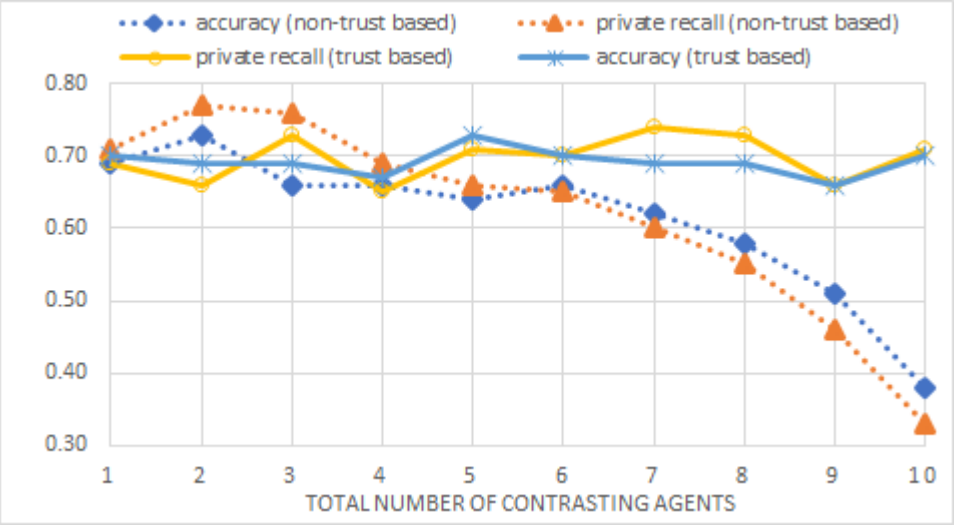

Figure 4.4. Accuracy of an agent with 18 friends

For each iteration of different number of friends evaluation, the simulation picks random n friends of the agent for five times. It runs 10 times for the same group of friends and reports the average of the results of multiple runs. In total, the simulation runs 50 different times for n-friends. Since each agent in the environment has different relations with other agents, evaluation results for same group of agents might be dependent on data (agents and their relations). Therefore, we choose random n-friends for five times to make evaluations data independent. We also repeat the evaluation for the chosen contrasting agents because each simulation randomly distributes images. Finally, the accuracy values of estimation from external data are examined for the agent both in a trust-based and non-trust based environments. In Figure 4.4, we plot the accuracy and private recall values of both trust based and non-trust based environments. The x axis shows the number contrasting friends that the agent has. We see that as the number of agents that have contrasting views increases, the accuracy of the non-trust based approach decreases sharply. When more than half of the agents have contrasting values, this yields an accuracy of only 0.39. This is expected since the agent makes decisions based on other agents' whose privacy understanding is completely opposite of its own. However, when trust is employed, the agent can maintain an accuracy of 0.7, even when the number of contrasting agents is half of its friends.

Notice that the accuracy results presented here are lower than the previous parts because we present only the results of the external estimation not the full system. Also, the setup of the environment in this part is a bit different from previous scenarios. Since we want to observe how the trust implementation affects the estimation from external data, we increase the threshold value from 0.01 to 0.1 to make external estimation more active. This effect of the threshold is presented as the line of *internal/all* in Figure 4.2. The images that are directed to external estimation are between the threshold values and their confidence values are close to average (Equation 2.4). Therefore, they are more prone to be misclassified. The important point of the figures in this part is the difference between the results of the trust based and those of non-trust based environments.

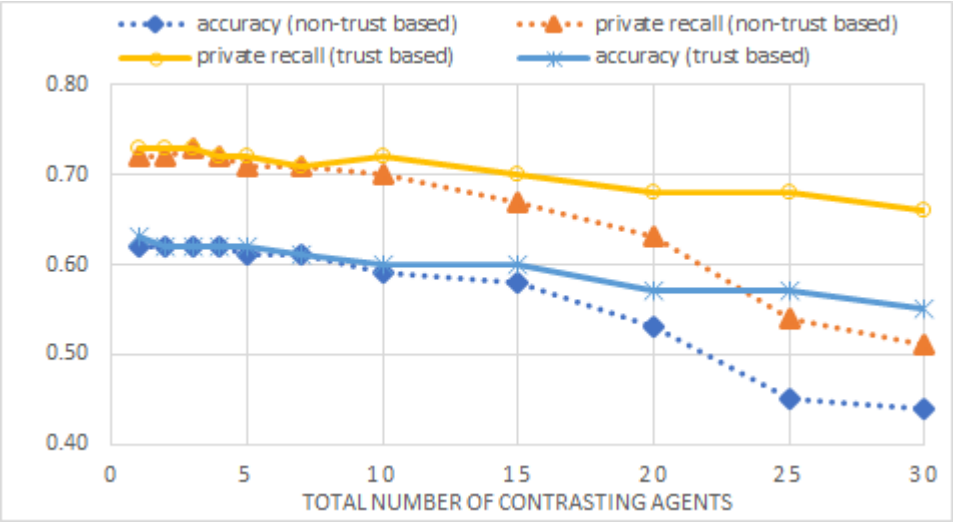

Figure 4.5. Average accuracy of all agents

We also study this setting from a macro view to understand the effect of contrasting agents to the whole network. We again vary the number of contrasting agents. Again, we chose random n agents from 52 different agents in the environment. The simulation is repeated 10 times for the chosen n-agents and we randomly choose set of n-agents for five times. Figure 4.5 shows that both private recall and accuracy drops considerably when trust is not employed in the system. These results are average results of the remaining agents, *e.g.* when there are total 10 contrasting agents, accuracy values correspond to the average accuracy of remaining 42 agents in the environment. External estimation accuracy of the agent and the overall system show that the trust

makes the system robust against agents with contrasting views. Increasing the number of contrasting agents in the network decreases the accuracy, but this decrease is much slower in a trust based environment compared to the non-trust based one.

4.2.4. Tag Analysis

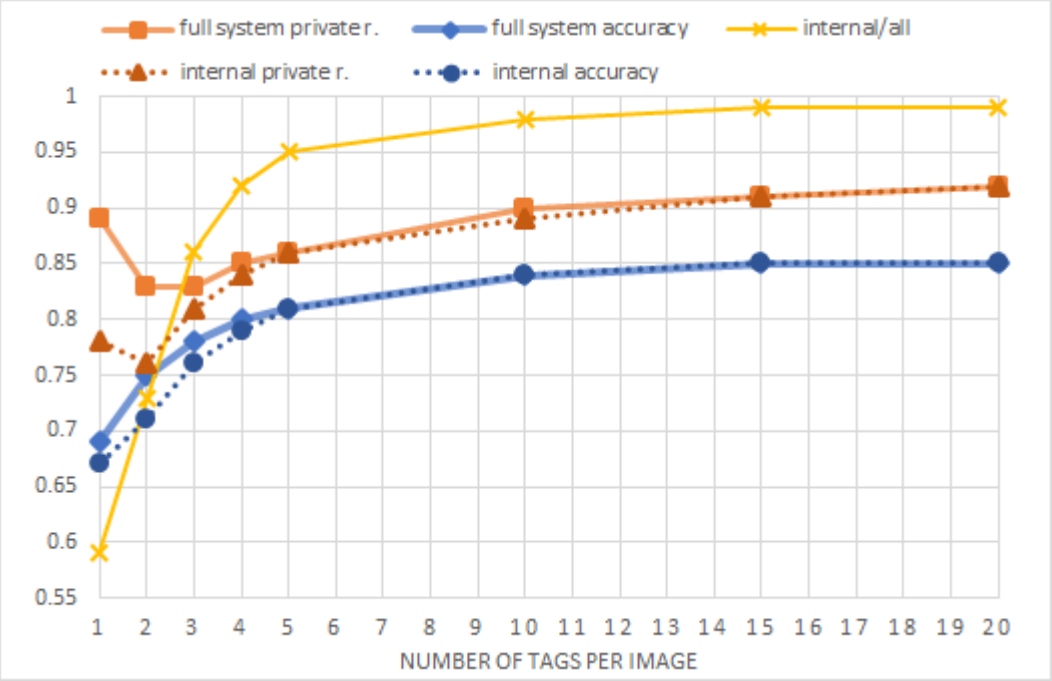

Figure 4.6. Accuracy against different number of tags per image

Our simulations use the tags generated by the general model of Clarifai [25]. For each image, Clarifai generates 20 different tags. In all the simulations up to now, we allow agents to use all 20 tags while estimating the privacy setting of an image. However, in various systems, an image might not be tagged automatically but might be tagged by a human instead, which would have fewer tags than 20. Or another tool, which creates fewer number of tags, might be used. Hence, PELTE should yield an acceptable accuracy even when an image has fewer tags. To study this, we first run the simulations where we vary the number of tags used per image and measure the accuracy of PELTE. We evaluate both the internal estimation and the full system, which uses both internal and external data, with 3000 training images and 1000 test images. In Figure 4.6, we present accuracy and private recall values for each experiment. Even though the public recall values are not included to make the lines clearly

distinguishable, they can be easily estimated from the presented values. The figure shows that having as few as five tags guarantees an accuracy more than 0.81 that is comparable to a case with 20 tags (0.85). In an extreme case, even when each image has only one tag, the system still achieves an acceptable accuracy of 0.7.

On the other hand, the full system results in Figure 4.6 show that having fewer tags pushes the system to use external data more. This effect can be seen from the line of *internal/all*. And the usage of external data in the case of few number of tags increases the system accuracy. We can conclude that the privacy estimation from external data improves the accuracy results not only for the cold start phase, but also for the systems having fewer number of tags.

Table 4.1. Frequencies of Top Five Tags

Private		Public	
Tag Name	Frequency	Tag Name	Frequency
people	0,93	no person	0,72
adult	0,73	outdoors	0,36
portrait	0,72	nature	0,33
woman	0,68	travel	0,31
man	0,55	people	0,28
Average	0,011	*Average*	0,007

Another important point to realize in the figures is the difference between private and public recall, where private recall is higher than public recall. To understand this difference, we study the distribution of tags in the dataset. Klemperer *et al.* [11] evaluate the accuracy of the tag based access policies by considering their complexity. The way they measure the complexity is that counting the number of unique tags in the access policy rules. They state that there is a trade-off between accuracy and the number of unique tags. Their results show many access policies are both simple and accurate. Similarly, we study the difference in the number of unique tags to see whether a correlation between our study and the study of Klemperer *et al.*. In our

dataset, public images have 2703 unique tags (0.77 per image), whereas private ones have 1718 unique tags (0.49 per image). While average occurrence of a tag is 25.9 for 3500 public images, that is 40.8 for the same number of private images. Table 4.1 shows the most frequently used five tags of private and public images in our data set. It can easily be seen that even for the most frequent tags, tags of private images are much more frequent than those of public images. For example, *people* tag has frequency of 0.93, whereas the top tag for public images is *no person* with a frequency of 0.72. Other tags of public images have considerably lower frequency value; e.g., *outdoors* 0.36. Because the tag *people* is almost in all private images, classification of those images is highly accurate even for the case of which only one tag is used. Average frequency values in the table show that it is more probable to see the same tags in different images of the private image set. We conclude that the gap between public and private recall values in the experiments arises from the difference in the number of unique tags associated with each category.

Another important feature of the tags is that private images are mostly related with human beings. On the other hand, public image contents are variations of nature, outdoor, travel, and so on. These features of the dataset and privacy labels of it are similar with the privacy object classes identified by the recent work, deep-multi task learning approach [12]. Their centralized classifier achieve its best performance, accuracy value is roughly 0.9, with a huge training data (900 image sets each one has 100 images). In our study, we obtain average accuracy of 0.85 after the agents in the distributed system have only 30 images.

4.2.5. Overcoming Cold-Start Problem

All previous simulation scenarios construct an OSN environment from scratch. As we explained in the dynamism feature of the OSNs, their network graphs change in time. More realistic scenario for the contribution of the estimation from external data can be that users in an OSN have many shared images and a new user joins the system. For example, a user creates a new account in the OSN and adds some other users, who have already shared images in their profile, as friends before sharing an image.

Table 4.2. Confusion matrix for a new comer

		Estimated Labels			
		Internal		Full System	
		Private	Public	Private	Public
Actual Labels	Private	102	26	95	11
	Public	52	65	35	119
Metrics	Accuracy	0.68		0.82	
	Private R.	0.80		0.90	
	Public R.	0.56		0.77	

To evaluate the scenario of new comer, a new user is added to the system after 3000 images are already shared in the environment. Since a user can see the previously shared images of a new friend, the new user adds data of her friends' shared images to her external tag table. The new user shares approximately 10 images at each simulation run. Table 4.2 shows the user's confusion matrix after 30 simulations runs. In the internal part of the confusion matrix, the user decides how to share an image herself by using only the estimation from internal data property of the system. In the full system part, estimation from external data is also involved. Rows of the table correspond to the actual labels of the images and the columns correspond the what is estimated by PELTE for the images. Because images are randomly shared in the simulations, the total number of images in the table are not equal. Private recall, public recall, and accuracy values are calculated for both internal and full system scenarios and presented in the table. Since the user only shares 10 images at each run, accuracy of the estimation from internal data is 0.68. On the other hand, accuracy becomes 0.82 when the system involves external data of the user to the estimation process. It is obvious that the using external data improves the accuracy results for new users.

4.3. ReBAC Experiments

The experiments in Section 4.2 show how PELTE meets the principles introduced in Chapter 1. But the data limit these experiments to a network type where only one relationship type is used. To show that PELTE can work on OSNs where ReBac is implemented, we synthetically extend the data and analyze its success by doing new experiments on the data.

4.3.1. Dataset Creation

We start by creating a new network graph from the network data we have. In the real network dataset, we have only the friend relationship type. We randomly change the relation type between agents into three types: Friend, Colleague and Family. Since PELTE supports only bidirectional relations, we ensure that the relationship type from agent a to agent b is the same type with the relationship type from agent b to agent a.

Then we continue with extending image dataset. Firstly, we generate privacy labels of each image against each relation type. We run the system on random privacy labels and these experiments simply show that the system can work on ReBAC. However, accuracy results do not mean anything since the data are randomly generated. Therefore, we study the tags to label images in detail.

The solution we find is to create tag based characters on agents. That is, an agent character considers a group of tags as private and another group of tags as public. However, creating random groups of tags would not yield to results that we expect. The possible problem is that if we label tags randomly, tags that mostly co-occur may have different labels. Therefore, we create a co-occurrence matrix of the most frequent tags to find tag groups that contain more related tags in it. Co-occurrence matrix of the most frequent 20 tags is presented in Figure 4.7.

#	no person	people	outdoors	one	adult	nature	travel	portrait	woman	man	landscape	desktop	light	indoors	sky	wear	business	art	wood	water	winter	city	street	color	girl
people	0,29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
outdoors	0,81	0,31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
one	0,40	0,79	0,26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
adult	0,14	0,99	0,19	0,65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
nature	0,82	0,15	0,68	0,20	0,06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
travel	0,81	0,31	0,66	0,14	0,12	0,33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
portrait	0,19	0,84	0,19	0,71	0,70	0,14	0,05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
woman	0,12	0,95	0,18	0,58	0,80	0,09	0,09	0,66	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
man	0,15	0,98	0,19	0,56	0,85	0,06	0,12	0,56	0,65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
landscape	0,79	0,25	0,68	0,11	0,11	0,60	0,59	0,06	0,05	0,08	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
desktop	0,72	0,21	0,15	0,17	0,09	0,30	0,09	0,09	0,10	0,10	0,08	0	0	0	0	0	0	0	0	0	0	0	0	0	0
light	0,69	0,42	0,38	0,24	0,21	0,29	0,44	0,16	0,15	0,16	0,41	0,15	0	0	0	0	0	0	0	0	0	0	0	0	0
indoors	0,48	0,72	0,08	0,53	0,50	0,03	0,15	0,43	0,45	0,39	0,03	0,09	0,21	0	0	0	0	0	0	0	0	0	0	0	0
sky	0,86	0,14	0,74	0,07	0,04	0,54	0,74	0,02	0,03	0,04	0,64	0,07	0,38	0,01	0	0	0	0	0	0	0	0	0	0	0
wear	0,23	0,89	0,15	0,64	0,78	0,03	0,08	0,62	0,66	0,63	0,04	0,13	0,10	0,34	0,01	0	0	0	0	0	0	0	0	0	0
business	0,63	0,48	0,18	0,24	0,29	0,03	0,29	0,18	0,22	0,28	0,05	0,26	0,18	0,38	0,11	0,15	0	0	0	0	0	0	0	0	0
art	0,62	0,46	0,14	0,30	0,24	0,18	0,19	0,21	0,23	0,24	0,07	0,50	0,24	0,12	0,09	0,17	0,12	0	0	0	0	0	0	0	0
wood	0,89	0,19	0,57	0,21	0,07	0,57	0,28	0,08	0,05	0,05	0,43	0,21	0,21	0,19	0,19	0,05	0,09	0,10	0	0	0	0	0	0	0
water	0,83	0,20	0,69	0,14	0,08	0,59	0,72	0,04	0,06	0,06	0,63	0,08	0,31	0,01	0,59	0,02	0,04	0,07	0,18	0	0	0	0	0	0
winter	0,76	0,31	0,60	0,23	0,14	0,53	0,32	0,15	0,12	0,10	0,54	0,14	0,26	0,04	0,30	0,12	0,03	0,09	0,39	0,27	0	0	0	0	0
city	0,66	0,46	0,49	0,12	0,21	0,06	0,71	0,10	0,16	0,19	0,24	0,04	0,40	0,13	0,37	0,09	0,29	0,13	0,05	0,21	0,11	0	0	0	0
street	0,53	0,60	0,42	0,26	0,34	0,07	0,52	0,22	0,27	0,31	0,18	0,06	0,35	0,12	0,19	0,18	0,20	0,14	0,08	0,10	0,14	0,63	0	0	0
color	0,78	0,19	0,29	0,20	0,07	0,50	0,10	0,08	0,09	0,06	0,11	0,61	0,19	0,08	0,08	0,14	0,09	0,37	0,19	0,08	0,12	0,04	0,06	0	0
girl	0,07	0,94	0,19	0,68	0,73	0,12	0,07	0,82	0,87	0,40	0,06	0,09	0,15	0,32	0,03	0,39	0,07	0,16	0,04	0,04	0,09	0,06	0,12	0,07	0

Figure 4.7. Co-occurrence matrix of most frequent 25 tags

The algorithm of group creation is presented in Figure 4.8. We create tag groups as follows: starting from the most frequent tags, we create a group for each tag that does not have high co-occurrence ratio with any of previous tags. We use a threshold (line 10) to decide whether co-occurrence ratio is high enough or not. If a tag has co-occurrence ratio more than the threshold value with more than one tag, then it goes into the group of the tag with the highest co-occurrence ratio (line 19). If there is no tag group that the co-occurrence ratio of the group leader and the tag is higher than the threshold value, the tag becomes a group leader and we create a new group for the tag (line 16). Finally, we distribute small groups into other groups to make the created groups bigger. While distributing the tags, we follow reverse order on tags according to their frequency. Thus, we remove the groups of tags that are less frequent and enlarge the groups of tags that are more frequent. Tags of the removed groups goes to the group that co-occurs most with the tag. This is the same group finding operation as we do in previous steps (line 9). The difference is that we do not apply any threshold while finding a new group to distributed tags. These operations iteratively continue until there is not any group smaller than the count limit.

We group the most frequent 100 tags as an example. First, we find the maximum co-occurrence ratio of each tag with other tags. Then, we calculate the average of maximum co-occurrence ratio of the tags. Since it is equal to 0.80, we choose the

Algorithm 4: CreateGroups

Input : T, array of list of tags in each image

Input : n, number of tags will be used in groups

Input : min, minimum value to be assigned to a group

Input : $size$, minimum size to form a group

Output: G, tag groups of most frequent n tags

```
1  CreateGroups (T, n, min, size)
2      tagList ← getMostFrequentTags(T, n);
3      coMatrix ← calculateCooccurrenceMatrix(T, tagList);
4      G ← initMap();
5      G ← G.put(tagList[0], initList());
6      foreach tag t1 in tagList do
7          value ← 0;
8          leader ← null;
9          foreach tag t2 in G.getKeys() do
10             if (coMatrix[t1][t2] > min) and (coMatrix[t1][t2] > value) then
11                 value = coMatrix[t1][t2];
12                 leader = t2;
13             end
14         end
15         if value == 0 then
16             G ← G.put(t1, initList());
17         end
18         else
19             G.get(t2).add(t1);
20         end
21     end
22     distributeSmallGroups(G, size);
23 return G
```

Figure 4.8. Algorithm of Group Creation

parameter of minimum co-occurrence ratio as 0.80. To be sure that there is not any group without a member other than the group leader, we set the parameter of the minimum number of group members to one. The resulting tag groups are presented in Table 4.3. The first column is the group leader and the second one is the list of tags that are in the group of the tag leader in the same row. Additionally, we presented minimum and average co-occurrence ratio between the tags of the group and the group leader.

In Figure 4.7, we can see that "no person" and "people" are the most frequent tags of the dataset. Furthermore, they have high co-occurrence ratios with other tags. When we distribute the tags that do not have a group member, they mostly go into the group of "no person" and "people" tags. Thus, their groups become highly dominant between the other groups. Since their groups include most of the tags, we remove "no person" and "people" from the list of tags and re-group the tags. Our aim is to create more scattered tag groups. Again, we find the maximum co-occurrence ratio of each tag. This time, the average of them becomes 0.69. We set the threshold parameter as 0.69 and run the same group creation process again. The result is presented in Table 4.4. It is obvious that the tags are separated more with respect to the first groups. Hence, we have more tag groups. We use the tag groups in Table 4.4 to evaluate the performance of PELTE in an environment in which ReBAC is implemented.

4.3.2. Performance

We do experiments on ReBAC by creating various agent characters that use tags for privacy decision. A character can have two lists: private tags and public tags. These lists are specific to the relationship types. Whenever an agent shares an image, it checks whether the tags of the image are in its tag lists for each relation type. If there are more private tags than the public ones, then the agent does not share the image with the corresponding type of relationship or vice versa. Contrarily, the agent does not share the image with the relationship if there are more public tags than private tags. Moreover, in the case of equality or there is no tag is in either lists, the agent permits the access of the users with that relationship type.

Table 4.3. Tag groups of the most frequent 100 tags

leader	group members	avg ratio	min ratio
no person	outdoors, nature, travel, sky, light, evening, wood, water, architecture, tree, building, daylight, dawn, sunset, food, animal, house, dark, leaf, bright, flower, flora, environment, fair weather, still life, closeup, decoration, retro, summer, window, color, industry, mammal, business, paper, park, landscape, old, blur, reflection, sun, technology, vintage	0.798	0.574
people	adult, portrait, woman, man, wear, two, girl, child, music, recreation, group, festival, fun, competition, model, facial expression, face, beautiful, monochrome, family, action, religion, education, fashion, young, love	0.829	0.546
one	cute, eye, little	0.637	0.594
desktop	abstract, texture, pattern, design, symbol	0.821	0.724
indoors	room, furniture	0.882	0.877
art	illustration	0.670	0.670
winter	snow, cold, weather, ice	0.926	0.840
city	urban	0.860	0.860
street	road	0.681	0.681
vehicle	transportation system	0.776	0.776
sea	ocean, beach	0.886	0.796

Table 4.4. Tag groups of the most frequent 100 tags without people and no person

leader	group members	avg ratio	min ratio
outdoors	sky, tree, daylight, park, environment, house, wood	0.731	0.547
one	portrait, face, monochrome, still life, young	0.609	0.465
adult	woman, man, wear, girl, music, group, two, facial expression, competition, festival, recreation, education, love	0.706	0.383
nature	summer, leaf, flower, flora, fair weather, beautiful	0.828	0.498
travel	water, city, architecture, building, sea, ocean, religion, reflection	0.684	0.514
landscape	dawn, sunset, sun, beach	0.769	0.717
desktop	design, abstract, decoration, symbol, texture, pattern, food, paper	0.719	0.353
light	evening, dark, blur	0.588	0.489
indoors	room, furniture, family, window	0.711	0.507
business	industry, technology	0.588	0.493
art	retro, illustration, old, vintage	0.558	0.446
winter	snow, cold, weather, ice	0.926	0.840
street	urban, road	0.693	0.681
color	closeup, bright	0.691	0.683
fashion	model	0.869	0.869
child	fun	0.585	0.585
vehicle	transportation system, action	0.624	0.472
cute	little, eye	0.718	0.700
animal	mammal	0.772	0.772

In the simulations, we use real data labels that come from the dataset as the decisions of agents against the *Friend* relationship type. For the *Colleague* and *Family* relationship types, agents use the tags of images for privacy decisions as we explain above. Characters are randomly assigned to agents. We use 4000 images as training data and 3000 images as test data. Threshold θ is 0.01 and simulation runs 10 times. In the training phase, agents learn from the decisions that they make based on the tag list in their characters. In the test phase, they compare the estimated privacy setting with the privacy decision based on the tags. We put the following scenario into practice:

- In the first experiment, we have just one character type and all the agents decide according to the same character. This character mimics an architect who goes hiking on weekends. She does not want to share her images taken in nature during hiking with her colleagues. Moreover, she does not want to share her images related to her job with her family since these images are irrelevant to her family. We implement this character by using the tags in the group of "nature" in Table 4.4. We consider these tags are private against Colleague relationship and we put them into the private tag list of the relationship. Additionally, we use the tags in the group of "business" as private for Family relationship. We run the simulation of the scenario and obtain the results in the first three rows of Table 4.5. Public recall, private recall, and accuracy values are given separately for each type of relationship. PELTE obtains 0.76 accuracy value in estimating privacy settings for Colleague relationship. Similarly, accuracy for Family relationship is 0.72. We see that the given tag groups create meaningful privacy preferences on the character. However, friend relationship uses the real data and has the best results. We conclude that PELTE can estimate privacy settings of each relationship in the degree to which privacy preferences of users are consistent and clear.
- In the second experiment, we enhance the same character to get better results. We add tags to the public tag lists of each relationship. In the first experiment, more images might be considered as private because only private tags are added

as the user preference. Because she is an architect, she wants to share art style images taken in streets with her colleagues. Moreover, she wants to permit her family members to access the images taken in a home environment. We mimic these behaviors by adding the tag groups of "art" and "street" in Table 4.4 into the public tags list of the *Colleague* relationship type. Also, we add the group of "indoors" into the same list of the *Family* relationship. We run the simulation and we get the results in Table 4.5 with experiment number two. Since the user now has more precise privacy preferences on images, we observe a minor improvement on the accuracy results of the second experiment for both *Colleague* and *Family* relationships.

- In the last experiment, we add one more character into the environment. Its privacy preferences are completely opposite of those of the first character on the same contexts against colleagues, *i.e.*, the public tags of the first character become the private tags of the new character and vice versa. However, their preferences are neither similar nor opposite against the family relationship. The second character considers images having the tags in the group of "one" as private. The results in the last three rows of the Table 4.5 show that the accuracy of Colleague relationship decreases. Since the agents with opposite privacy preferences join the system, they probably affect each other negatively and decrease the accuracy. However, the decrease in the accuracy value is not too much. As we see in Section 4.2.3, trust implementation of PELTE makes the system robust against the contrary preferences. On the other hand, the results of the *Family* relationship is much better than those of previous experiments. This might be a result of using the tags in the group of "one". Since it is one of the most frequent tags in the list, agents become more consistent on their privacy preferences. Note that, these are the overall results presenting the sum of results that comes from all the agents in the environment even if they have different characters.

These results show that given set of tags can help agents specify their attitudes against images in specific contexts. Even though our aim is just to create synthetic data to evaluate performance of PELTE in environments using ReBAC, we obtain

Table 4.5. Performance of PELTE on ReBAC experiments

Exp. no	Relationship Type	Private Rec.	Public Rec.	Accuracy
1	Friend	0.93	0.76	0.85
1	Colleague	0.79	0.76	0.77
1	Family	0.80	0.71	0.72
2	Friend	0.94	0.76	0.85
2	Colleague	0.82	0.75	0.78
2	Family	0.74	0.74	0.74
3	Friend	0.94	0.76	0.85
3	Colleague	0.75	0.74	0.75
3	Family	0.87	0.73	0.77

meaningful results from tag based agent characters. We create the character of an imaginary person and estimate the privacy settings of the agents having the specified character. We see that PELTE is able to store the data coming from different types of relationship and estimate the sharing actions of agents against each relationship type successfully. Furthermore, privacy estimation from initially given set of tags when there are not available data might be a future direction for PELTE. Initial sets of tags might be created automatically by analyzing the data of images in the social network as we did in Figure 4.8. And then users may be asked to label the tag groups. Thus, PELTE can learn privacy preferences directly from users even before they share images. We want to emphasize that the aim of the ReBAC experiments we do in this section is not to improve the results the accuracy results we obtain from real data. Since the real labels of the dataset correspond to more consistent privacy preferences, creating synthetic data is not expected to be more successful with regards to privacy estimation. Nevertheless, we evaluate PELTE for ReBAC on data that are created logically instead of random creation and PELTE successfully estimates the privacy settings of the images.

5. DISCUSSION

We propose an agent based approach for images on OSNs to preserve users' privacy. Agents store the tags of uploaded images and then use these tags to automatically estimate privacy settings of a new uploaded image. We implement the approach and test it on a simulation environment. We develop a simulation environment on which we can test the performance of our approach. The environment allows various number of agents to exist and estimate privacy settings at the same time. The agents' privacy is compliant with ReBAC. The tags of the images are created automatically via a tool. We create synthetic data from real data to test it on relation based privacy settings. Results show that proposed model can estimate privacy setting of images accurately even on OSNs where more than one relationship type is supported. Even though the simulations that we do are performed in relatively small social network, because our approach is agent based and that the computations are done locally, it is expected to work on a large network as well. In the same scenarios that we evaluate PELTE, alternative approaches that use machine learning technique would require initial set of tags to represent input space as a vector. However, we assume that the system starts from scratch and initially it knows nothing. Thus, PELTE addresses the dynamism property of the OSNs, which has to be ignored by machine learning techniques. PELTE is accurate even when the available data are limited. Further, it can help new users or users who have shared few images before by suggesting privacy settings based on the data that comes from privacy settings of their friends' images.

There are different ways of considering how privacy can be preserved in OSNs. Some of the recent literature analyze information disclosure to figure out possible ways of privacy breaches [30, 31]. Several other approaches consider how privacy violations can be detected [32, 33]. These approaches help users after a privacy violation takes place. Another set of approaches consider how entities can resolve privacy conflicts among themselves. They employ techniques like collaborative access policy administration [34,35], argumentation [17], negotiations [36,37], help of a mediator [38], secret key sharing [39] and so on. Yet, another set of approaches study how automated sys-

tems can help users manage their privacy by suggesting privacy settings or filtering information when needed. This last set is in line with our research question. We discuss some of these approaches next.

Squicciarini *et al.* [13] propose an Adaptive Policy Prediction (*A3P*) system to help users in specifying privacy preferences for an uploaded photo. It automatically produces a privacy policy for an image by analyzing user's criteria which influence her privacy preferences. *A3P* has two components called *A3P-Core* and *A3P-Social*. *A3P-Core* is the main component that uses user's previous policies to find an appropriate privacy policy for an uploaded image. In some cases, it invokes *A3P-Social* to find a privacy policy. This occurs when a user does not have enough historical data or there is a major change in user's social network. While *A3P-Core* finds an appropriate privacy policy for an uploaded image via using the user's previous policies, *A3P-Social* tries to find a privacy policy from another user, who has similar social context and strictness level with the user. *A3P-Social* component of this work is similar to the external estimation part of our proposed model. *A3P-Social* is basically responsible for inferring privacy policies for the user by using other users' data instead of user's own data. It finds someone from a group of users, who have common values for attributes in user profiles. However, *A3P-Social* finds a privacy policy from the entire network. The underlying assumption here is that an entity would have access to the entire network, which is in itself a violation of privacy. To deal with this, we have designed our system such that each agent can only see the privacy settings of the images that were already shared with it. Further, since *A3P-Social* is not implemented or evaluated, its success is not known. In our work, agents use the data of shared images from friends in the degree of trust against them to estimate privacy settings externally. We have quantitatively shown that PELTE is successful. Our evaluations on the simulation environment show that PELTE is accurate when the variation of the data is limited with image tags or even when the number of images and tags are few.

Albertini *et al.* [40] have developed a system, which aims to recommend privacy policies that are similar to users' subjective privacy decisions. A user's previous privacy policies are stored in the itemset catalogs as a set of object descriptors. The impact

of resource's features and the characteristics, e.g., relationship type and trust value between users, of a requester on privacy preferences are taken into consideration in the model. Learning of privacy preferences is done by two procedures: frequent itemset lookup and association rules extraction. The extracted rules are combined to recommend a privacy policy. Because contents of resources and the sign of privacy rules are stored separately, combination process of extracted rules is vague. We do not know how exactly the system generates a privacy policy. For the evaluation, they construct a small network graph with 33 nodes and 52 edges. Only the ratio of recommended privacy policies is presented in the results. The accuracy result of the recommended policies is missing. Our simulation environment uses a bigger network and real labeled image dataset. Also, the accuracy of the predicted privacy settings is presented to show that our proposed model is both applicable and successful. Moreover, cold start problem of the recommender system in their work is left as future work. In PELTE, external estimation of privacy settings addresses the cold start problem.

Squicciarini *et al.* [14] explore users' privacy preferences for uploaded images on online social networks. They analyze visual features (SIFT, edge direction, facial detection, RGB, sentiment) that are assumed to be informative in privacy of images. They use both visual features and tags of images as input parameters to classify images. Different learning models, e.g. Naive Bayes, k-nearest neighbors, support vector machine, etc., are tested on the image dataset. Since the images are not linked to individual users in classification phase, the proposed approach becomes a centralized system. Therefore, their approach does not consider privacy preferences individually. Even though training of their model for each user separately is technically possible, it needs dataset of more than 3000 images as a batch to reach maximum accuracy. They obtain accuracy of 86.5% when 3600 images are used for training. It is not practical to assume that an OSN user can have that many images on her account. On the other hand, we deal with limited data by using agent based approach instead of a centralized one. Moreover, they find out that tags are more informative than visual features. Therefore, we also use content tags in our proposed approach.

Klemperer *et al.* [11] study how people can use tags for access control in photo sharing. They make a laboratory study and recruit 18 participants who take at least 100 photos per year and have tagged photos OSNs other than Facebook and Flickr. The participants upload 40 photos they had previously tagged. The study consists of three phases. The first phase is organizational tagging. After original tags are stripped from the photos, they ask the participants to re-tag their photos with the objective of finding photos more easily in the future. They collect participants' privacy preferences. Participants select a preference option for each pair of photo and friend (also public). Preference options are strong allow, weak allow, strong deny, weak deny, and neutral. This data are used as a ground truth for the evaluation. They apply the machine generated rule sets to the participants' photos and show example rules to the participants in order to make them familiarized with the idea of tag based access control rules. In the second phase, they ask the participants to add or delete from the tags. Then, machine-generated rules are applied again. In the last phase, misclassifications are presented to user for the last refinement of tag. All results and tag deletion/additions are collected. Because the rules generated from organizational tags conflict for 7.8%, they conclude that organizational tags can express many access policies. Similarity between organizational and original tags is analyzed by comparing overall rate of conflicts when using original and organizational tags for rules. to the results, the tags that are created in organizational tagging phase are not very different from the tags the participants normally create. Finally, the effect of tag addition/deletion in the second and third phases is analyzed. In the second phase, mostly new tags are added and results are improved 2.7%. In the third phase, the results are improved 1.1% by the participants' fewer operations. It shows that when users understand how tags affect access-control, they can use tags more effectively.

Our work is inspired from the study of Klemperer *et al.* [11] in the sense that the image tags are successful enough to estimate privacy setting of images even users do not think tags will be used for this purpose. In addition to that, users can improve their tagging approach if the estimated privacy setting do not fit their privacy needs. Since PELTE updates tag tables for every image, the idea of improvement in tagging is supported by our dynamism principle. Moreover, they state that the participants

are more concerned with false allows than false denies. Similarly, the results show that it PELTE is better at private estimation than public one.

Zhong *et al.* [15] propose a personalized model to classify images. Their main challenge is that the limited user data are not enough to train a personal classifier accurately. Therefore, authors propose a model based on privacy groups, which are subsets of users. The model learns these privacy groups and associates a new user with the groups. Their method processes an image into patches to find spatially localized regions, which may contain a sensitive content. An image is considered as a private image if it has at least one region with a private content. They calculate a $0-1$ latent variable, which indicates a user's belongingness to a group, for each user by using a user's privacy labels to image patches and her profile data. Expectation-Maximization algorithm is used to find whether an image is private or public for a user. They model user profiles as a 30-dimensional binary vector, which corresponds to cardinality of seven demographic variables. When a new user does not have any labeled image, the system finds her group, based on the profile data. They evaluate the system on randomly sampled 2700 images from the Picalert dataset. Dataset is split into 90 subsets and each subset is assigned to two Mechanical Turk workers. Workers' tasks are labeling images and providing demographic data about themselves. The system run with following parameters: 40 content types, six user groups, and 100 image patches. Because EM may not converge properly, initial values are provided to the system via running it with seed users in only one group. In the image prediction evaluations, the whole system is compared with the baselines, which are generated by subtracting different properties of the system. The proposed model gets the highest, 79.31%, accuracy value. Also, it is shown that even there are insufficient amount of data provided by a user, the system can predict the user's privacy label for an image more accurately than a general (non-personalized, centralized) model. PELTE is similar to their approach in the sense that PELTE predicts a new user's privacy preferences via other users' privacy preferences. However, they relate users with each other by using their profile data. Instead of using extra data, we use data that come from shared images and relate people according to similarity of their privacy understanding. Additionally, the complexity of their approach grows with the number of users, it is not

possible to execute their personalized model on agent based environment, where high computation power is not available. PELTE obtains high accuracy while preserving simplicity principle. The number of privacy groups and those of private contents are chosen with a pre-processing on the dataset. It is still an open question how to optimize these numbers in a running system. Another issue is that new users are not integrated into the training part (defining group properties, which are used in EM algorithm) of the system. Our dynamism principle let new users join the system and enrich their own personal tag tables over time.

Kepez and Yolum [18] propose a framework in which agents can learn users' privacy concerns regarding posts. If a user has shared many posts before, the user's agent uses shared posts as a training set for machine learning techniques. If the training set is not big enough, accuracy of classifiers decreases. In this case, authors come up with a multi-agent solution. An agent asks all the agents for the sharing decision. Then, it calculates a weighted average of the all responses for a final decision. The weights are the trust values against other agents, which is calculated by asking agents for the posts that are already shared by the user. If other agents return the same decision with the user, this reflects positively to their trust values. While that work focuses on posts, our proposed model works on images. In multi-agent part of their work, an agent asks all agents for the decision without analyzing their knowledge about the content of the new uploaded post. An agent only takes into account whether the other agent would decide the same sharing constraints for a previously shared post. In our proposed model the aim is that acquiring knowledge from other users, who have already shared images with similar content.

Fang *et al.* [41] propose an approach based on active learning to help users control their friends' accesses. It uses specified privacy preferences as an input to predict privacy settings of unspecified ones automatically. Wizards collect user's privacy preferences by asking privacy settings for item and friend pairs. An algorithm based on edge-betweenness is used to find densely connected nodes, which are called communities, in the user's network. Users' extracted features and communities are used for uncertainty sampling to find most informative friends. Since the user may give up

any time while asking users privacy settings of item and friend pairs, they are ordered according to informativeness values. A user's labels of her friends are used as inputs by the classifier to label unlabeled friends. The classifier constructed for prediction by using labeled data is a decision tree. Since the learning model is based on privacy preferences against friends, the system can only specify a general privacy settings. It cannot generate item based privacy settings for different contents, e.g. photos. Also, their system cannot create a successful classifier if the user does not label sufficient amount of data. On the other hand, PELTE benefits from friends' data to handle the issue.

Kökciyan and Yolum [32] propose a semantic approach to detect privacy violations in OSNs. Three main contributions of the work are meta-model to represent online social networks formally, semantic approach, which uses description logic to conform meta-model, and ontology based software tool of the proposed model. Privacy requirements are defined as commitments between two agents in an agent based social network. The purpose of the system is to detect if there is a violation of commitment, which means a privacy breach. Since the proposed system uses a semantic approach, it can infer implicit privacy violations. Their algorithm for detection is both sound and complete, but privacy policies are manually specified by users. Our proposed model can generate privacy policies automatically. In this regard, our work will complement the work of Kökciyan and Yolum.

Vanetti *et al.* [42] propose a text-based filtering mechanism, called Filtered Wall, to filter unwanted messages from OSN user walls. The mechanism is customizable according to user's privacy preferences. Radial Basis Function Network is chosen as a machine learning model to classify short texts. Messages are represented by bag of words in the texts, document properties and contextual features. The short text classifier automatically decides whether the message will be published or filtered. Their content based learning approach deals with only messages. In our work, we perform a content based approach for images. Furthermore, their system allows users to define filtering rules and manage blacklists. We implemented a similar method for images in a way that users can give an initial set of tags to share specific contexts either privately

or publicly.

Akçora *et al.* [43] propose a model based on the concept of risk which a user takes when interacting with other users in OSNs. The proposed risk measure is basically related with users, subjective risk perception, and risk judgment. Because users' risk attitude may vary, the system asks for owner feedback to estimate risk. For the risk judgment, the authors take into account similarity of users and benefits. Benefits can be thought as new information which encourage users to interact with users who are not similar. In the risk learning phase, supervised learning techniques are used. Because social graph can change fast, active learning is preferred instead of learning from fixed training set. The owner is repeatedly queried for labels of selected unlabeled strangers, who are clustered according to two dimensions. The first one is the network similarity, which measures both mutual friends and the connections among mutual friends. The second dimension is profile similarity which is used to refine network similarity groups. The classifier of the system uses a graph based approach where strangers are represented as nodes and edges weights are based on their profile similarities. The classifier predicts similar labels for similar neighbors on the graph, by exploiting the random walk strategy. The active learning phase terminates when a high accuracy, which is related with a confidence value specified by the owner, is reached. In their experiments, Facebook data is collected via Facebook application, called Sight. The dataset includes information about 47 Facebook users, $172,091$ stranger profiles and $4,013$ owner-defined risk labels. For each user, six network similarity groups are created. In the risk label prediction evaluation part, the labeling stops if there is no classification change for at least two rounds. For average 80% confidence, the label prediction stabilizes in about three rounds. $83,36\%$ of the predicted labels exactly match the owner labels. A comparison between network and profile similarity based pools and only network similarity based pools shows that using them together improves the results. The results show that photos are the most important benefit items for the users. Their approach is a good way to analyze risk in social network via a semi-automated privacy tool. The results they have obtained about benefit items are supportive for PELTE. Since photos are the most important benefit items and the most private ones for users, privacy of photos is an important issue for OSN users. Therefore, we specifically focus

on image privacy in this thesis. Additionally, our similarity based trust metric can feed their network similarity and risk analyses.

OSNs enable users to socialize and share content with other users. A user can upload an image in which other users also appear. Therefore, even a user pays attention to privacy, her friends may violate her privacy by sharing an image without her permission. In the recent literature, different approaches are proposed to resolve multi party access control of shared contents. Squicciarini *et al.* [44] examine privacy as a tax problem. They propose a collaborative management model based on Clark Tax algorithm. One of the points they emphasize as requirements of collaborative privacy management is automation to make process easier. PELTE can be used to assign bids to images automatically according to privacy value of the image. Similarly, our trust metric may estimate how much a user's friends will bid and then the user's agent can choose the best fitting bidding strategy against friends.

One future direction for the work can be that improving privacy policy definition. Such and Rovatsos [36] present a privacy policy definition based on relationship strength. Further, users can be given as a list to add either exceptional denies or permits on the privacy policy. We can extend PELTE in a similar direction to define more complex privacy policies. Thus, OSN users can (semi) automatically express more than public or private decisions against each relationship type thanks to PELTE.

Privacy in Internet of Things of is a newly growing study area of privacy [45]. The Internet of Things consists of smart devices that have an Internet access. People use various type of smart devices in daily life. Some of these devices can access to their personal data. Moreover, we inevitably exposure the devices that record voice, image, video, etc. Therefore, our private data becomes a part of the data stored in Internet of Things environments. Since the data collected by smart devices may violate privacy of people, the devices should take their actions regarding personal data more carefully. For this reason, we think that Internet of Things can be another future direction of PELTE. It can be integrated into Internet of Things environments after small customizations. Since it is both agent based and simple, it is capable to work in

smart devices, which have a low computation power and have a temporary connection to a centralized system. For example, surveillance devices may decide to whether share a scene with third party according to analysis done by our model. Thus, surveillance devices can be more sensitive against privacy of people.

REFERENCES

1. Ellison, N. B. *et al.*, "Social network sites: Definition, history, and scholarship", *Journal of Computer-Mediated Communication*, Vol. 13, No. 1, pp. 210–230, 2007.

2. Gürses, S. and C. Diaz, "Two tales of privacy in online social networks", *IEEE Security & Privacy*, Vol. 11, No. 3, pp. 29–37, 2013.

3. Such, J. M., A. Espinosa and A. García-Fornes, "A survey of privacy in multi-agent systems", *The Knowledge Engineering Review*, Vol. 29, No. 3, pp. 314–344, 2014.

4. Westin, A. F., "Privacy and freedom", *Washington and Lee Law Review*, Vol. 25, No. 1, p. 166, 1968.

5. Fong, P. W., "Relationship-based access control: protection model and policy language", *Proceedings of the First ACM Conference on Data and Application Security and Privacy*, pp. 191–202, ACM, 2011.

6. Gates, C., "Access Control Requirements for Web 2.0 Security and Privacy", *Proceedings of Workshop on Web 2.0 Security & Privacy*, 2007.

7. Strater, K. and H. Richter, "Examining privacy and disclosure in a social networking community", *Proceedings of the 3rd Symposium on Usable Privacy and Security (SOUPS 2007)*, pp. 157–158, 2007.

8. Sadeh, N., J. Hong, L. Cranor, I. Fette, P. Kelley, M. Prabaker and J. Rao, "Understanding and capturing people's privacy policies in a mobile social networking application", *Personal and Ubiquitous Computing*, Vol. 13, No. 6, pp. 401–412, 2009.

9. Lampinen, A., V. Lehtinen, A. Lehmuskallio and S. Tamminen, "We're in it together: interpersonal management of disclosure in social network services", *Pro-*

ceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 3217–3226, ACM, 2011.

10. Statista, *Number of monthly active Facebook users worldwide as of 1st quarter 2018*, `https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/`, accessed at May 2018.

11. Klemperer, P., Y. Liang, M. Mazurek, M. Sleeper, B. Ur, L. Bauer, L. F. Cranor, N. Gupta and M. Reiter, "Tag, you can see it!: Using tags for access control in photo sharing", *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 377–386, ACM, 2012.

12. Yu, J., B. Zhang, Z. Kuang, D. Lin and J. Fan, "iPrivacy: image privacy protection by identifying sensitive objects via deep multi-task learning", *IEEE Transactions on Information Forensics and Security*, Vol. 12, No. 5, pp. 1005–1016, 2017.

13. Squicciarini, A. C., D. Lin, S. Sundareswaran and J. Wede, "Privacy policy inference of user-uploaded images on content sharing sites", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 27, No. 1, pp. 193–206, 2015.

14. Squicciarini, A., C. Caragea and R. Balakavi, "Toward automated online photo privacy", *ACM Transactions on the Web (TWEB)*, Vol. 11, No. 1, p. 2, 2017.

15. Zhong, H., A. Squicciarini, D. Miller and C. Caragea, "A Group-Based Personalized Model for Image Privacy Classification and Labeling", *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*, pp. 3952–3958, 2017.

16. Zerr, S., S. Siersdorfer, J. Hare and E. Demidova, "Privacy-aware image classification and search", *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 35–44, ACM, 2012.

17. Kökciyan, N., N. Yaglikci and P. Yolum, "An Argumentation Approach for Resolv-

ing Privacy Disputes in Online Social Networks", *ACM Transactions on Internet Technology*, Vol. 17, No. 3, pp. 27:1–27:22, June 2017.

18. Kepez, B. and P. Yolum, "Learning privacy rules cooperatively in online social networks", *Proceedings of the 1st International Workshop on AI for Privacy and Security*, p. 3, ACM, 2016.

19. Bandura, A. and R. H. Walters, *Social learning theory*, Vol. 1, Prentice-hall Englewood Cliffs, NJ, 1977.

20. Burke, M., C. Marlow and T. Lento, "Feed me: motivating newcomer contribution in social network sites", *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 945–954, ACM, 2009.

21. Cutillo, L. A., R. Molva and T. Strufe, "Safebook: A privacy-preserving online social network leveraging on real-life trust", *IEEE Communications Magazine*, Vol. 47, No. 12, 2009.

22. Tsay-Vogel, M., J. Shanahan and N. Signorielli, "Social media cultivating perceptions of privacy: A 5-year analysis of privacy attitudes and self-disclosure behaviors among Facebook users", *New Media & Society*, Vol. 20, No. 1, pp. 141–161, 2018.

23. Huynh, T. D., N. R. Jennings and N. R. Shadbolt, "An integrated trust and reputation model for open multi-agent systems", *Autonomous Agents and Multi-Agent Systems*, Vol. 13, No. 2, pp. 119–154, 2006.

24. Wilson, R. E., S. D. Gosling and L. T. Graham, "A review of Facebook research in the social sciences", *Perspectives on Psychological Science*, Vol. 7, No. 3, pp. 203–220, 2012.

25. Clarifai, *General Model*, `https://www.clarifai.com/models/general-image-recognition-model`, accessed at April 2018.

26. Leskovec, J., *Stanford Network Analysis Project*, `http://snap.stanford.edu/data/egonets-Facebook.html`, accessed at April 2018.

27. PicAlert, *Privacy-Aware Image Tools*, `http://l3s.de/picalert/`, accessed at April 2018.

28. Li, K., Z. Lin and X. Wang, "An empirical analysis of users' privacy disclosure behaviors on social network sites", *Information & Management*, Vol. 52, No. 7, pp. 882–891, 2015.

29. Malhotra, N. K., S. S. Kim and J. Agarwal, "Internet users' information privacy concerns (IUIPC): The construct, the scale, and a causal model", *Information Systems Research*, Vol. 15, No. 4, pp. 336–355, 2004.

30. Krishnamurthy, B. and C. E. Wills, "On the leakage of personally identifiable information via online social networks", *Proceedings of the 2nd ACM workshop on Online Social Networks*, pp. 7–12, ACM, 2009.

31. Zhou, M. X., J. Nichols, T. Dignan, S. Lohr, J. Golbeck and J. W. Pennebaker, "Opportunities and risks of discovering personality traits from social media", *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, pp. 1081–1086, ACM, 2014.

32. Kökciyan, N. and P. Yolum, "PriGuard: A Semantic Approach to Detect Privacy Violations in Online Social Networks", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 28, No. 10, pp. 2724–2737, 2016.

33. Squicciarini, A. C., F. Paci and S. Sundareswaran, "PriMa: a comprehensive approach to privacy protection in social network sites", *Annals of Telecommunications-Annales Des Télécommunications*, Vol. 69, No. 1-2, pp. 21–36, 2014.

34. Carminati, B. and E. Ferrari, "Collaborative access control in on-line social networks", *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2011 7th International Conference on*, pp. 231–240, IEEE, 2011.

35. Hu, H., G.-J. Ahn and J. Jorgensen, "Multiparty access control for online social networks: model and mechanisms", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 25, No. 7, pp. 1614–1627, 2013.

36. Such, J. M. and M. Rovatsos, "Privacy policy negotiation in social media", *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, Vol. 11, No. 1, p. 4, 2016.

37. Keküllüoglu, D., N. Kökciyan and P. Yolum, "Preserving Privacy as Social Responsibility in Online Social Networks", *ACM Transactions on Internet Technology (TOIT)*, 2017.

38. Such, J. M. and N. Criado, "Resolving multi-party privacy conflicts in social media", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 28, No. 7, pp. 1851–1863, 2016.

39. Ilia, P., B. Carminati, E. Ferrari, P. Fragopoulou and S. Ioannidis, "SAMPAC: socially-aware collaborative multi-party access control", *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*, pp. 71–82, ACM, 2017.

40. Albertini, D. A., B. Carminati and E. Ferrari, "Privacy Settings Recommender for Online Social Network", *Collaboration and Internet Computing (CIC), 2016 IEEE 2nd International Conference on*, pp. 514–521, IEEE, 2016.

41. Fang, L. and K. LeFevre, "Privacy wizards for social networking sites", *Proceedings of the 19th International Conference on World Wide Web*, pp. 351–360, ACM, 2010.

42. Vanetti, M., E. Binaghi, E. Ferrari, B. Carminati and M. Carullo, "A system to filter unwanted messages from OSN user walls", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 25, No. 2, pp. 285–297, 2013.

43. Akcora, C., B. Carminati and E. Ferrari, "Privacy in social networks: How risky is your social graph?", *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pp. 9–19, IEEE, 2012.

44. Squicciarini, A. C., M. Shehab and F. Paci, "Collective privacy management in social networks", *Proceedings of the 18th International Conference on World Wide Web*, pp. 521–530, ACM, 2009.

45. Samani, A., H. H. Ghenniwa and A. Wahaishi, "Privacy in internet of things: A model and protection framework", *Procedia Computer Science*, Vol. 52, pp. 606–613, 2015.