Autonomous Agents and Multi-Agent Systems (2021) 35:7
https://doi.org/10.1007/s10458-020-09488-1

Assisting humans in privacy management: an agent-based approach

A. Can Kurtan[1] · **Pınar Yolum**[1]

Accepted: 1 December 2020 / Published online: 23 December 2020

Abstract
Image sharing is a service offered by many online social networks. In order to preserve privacy of images, users need to think through and specify a privacy setting 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, editing these privacy settings is cumbersome and requires too much effort, interfering with the quick sharing behavior expected on an online social network. Accordingly, this paper proposes a privacy recommendation model for images using tags and an agent that implements this, namely PELTE. Each user agent makes use of the privacy settings that its user have set for previous images to predict automatically the privacy setting for an image that is uploaded to be shared. When in doubt, the agent analyzes the sharing behavior of other users in the user's network to be able to recommend to its user about what should be considered as private. Contrary to existing approaches that assume all the images are available to a centralized model, PELTE is compatible to distributed environments since each agent accesses only the privacy settings of the images that the agent owner has shared or those that have been shared with the user. Our simulations on a real-life dataset shows that PELTE can accurately predict privacy settings even when a user has shared a few images with others, the images have only a few tags or the user's friends have varying privacy preferences.

Keywords Privacy · Online social network · Multiagent system

Preliminary ideas of this paper have appeared as an extended abstract in International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2018.

✉ A. Can Kurtan
a.c.kurtan@uu.nl

Pınar Yolum
p.yolum@uu.nl

[1] Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands

1 Introduction

Online social networks (OSNs) are web-based platforms where individuals interact with each other to share content [7]. While sharing content, an important concern of users is that the privacy of their content is preserved. Privacy in the context of OSNs can be understood in two main directions [24]. One perspective is that of *surveillance*. That is, users do not want their content to be used by the service providers to be profiled for marketing targeted goods, services or political opinions. Facebook-Cambridge Analytica data scandal [23] is a prime example, where Cambridge Analytica used millions of Facebook users' data without their consent for political advertising. Second perspective is that of *social*, where the users do not want their content to reach unintended users present in the network. We are interested in this second perspective of privacy, where we would like to support the users with the necessary tool to preserve their privacy as they share information online.

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 per image is also possible. 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. Moreover, many of the current OSNs are built with a centralized architecture, where the data are kept centrally. This means that these OSNs have the power to use the data for their purposes, ranging from profiling to targeting information. Even when they offer support for tasks, such as perserving privacy, it is not clear whom this task would serve, what data it would have access, and so on. A better way to approach this is to support individual users before they access the OSN, similar to the functioning of distributed OSNs such as diaspora* [14], where the individual data are kept on the user side. This has the advantage that personalized recommendations can be done to the user by considering her sharings and relations with others.

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 [21, 22]. A user can categorize her connections and specify fine-grained privacy settings in such a way that deciding for each relationship type [20, 43]. However, privacy settings are burden for many users because they find privacy settings difficult to manage and understand [20].

Various studies show that OSN users have even difficulties in understanding, let alone, setting the privacy settings of OSNs [45, 51]. Asking a user to manually set a privacy setting every time she is sharing an image will be time consuming and error prone [19]. The user will have to consider all the privacy implications of the image for various audience groups and then set the policy. More fundamentally, it is possible that the user does not know which privacy settings are appropriate for a content. This is especially true for the many new users in the system [37]. Recent studies show that users are in fact interested in using personal assistants to help them manage their privacy by providing notifications or

recommendations [11]. The aim of this paper is to devise agents that can help both experienced and new users of OSNs in recommending privacy settings for a new image that the user wants to share. The following criteria are important to understand the requirements of this recommendation task.

Personal data: While many existing OSNs are centralized in governance, an approach that helps users set their own privacy settings needs to use only the data available to each individual user, rather than using all of the data on the OSN. Having a central approach that works at the OSN level poses a threat to the system because the approach would be assumed to have access to the entire content shared on the OSN. Such approaches are prone to suffer from the surveillance problem mentioned before. Krasnova et al. [34] state that user privacy concerns mainly center on organizational risks such as collection and secondary use of their information. Once we assume that a central entity can access the contents of all users, potential use cases of such system exceeds the extent of privacy preserving mechanism and even it would jeopardize the privacy of the users, rather than helping them preserve their privacy. Hence, the privacy preserving mechanism should carry out the predictions for each user separately, using only the data that are available to the user herself.

Small data: Everyday millions of content are shared on one OSN. However, the number of contents shared by a single user is rather small. An approach that helps users set their privacy setting correctly needs to learn from this small data. This has two immediate consequences: (i) typical machine learning approaches that learn from big data well cannot be immediately used. (ii) If users do not have enough data to make reliable estimations themselves, a *cold start problem* would emerge. Thus, ideally we are in need of an approach that can use small data to make correct recommendations to the user.

Privacy variance: Definition of privacy is subject to personal understanding of each user [32]. While a user might not want her home pictures to be shared with colleagues, another user might be happy to share them with everyone. Thus, the proposed automated approach should predict the privacy of a content for a given user based on the expectations of the user. This means that even when other users in the system have different or contradicting privacy preferences, the approach should still recommend the right privacy settings to the user.

Robustness: OSNs differ in their size, user representation, or the content they allow. If a prediction algorithm that uses such information to make a decision, then it needs to be customized for each OSN or maybe even for each user before use. This jeopardizes the applicability of the approach. Ideally, the approach should work without preprocessing, customization, or configuration, so that it is generic enough to be applied in various OSNs. More importantly, the approach should work even when the previously seen content has missing information or even inconsistencies.

Although the problem of privacy setting recommendation for images have been extensively studied before, none of the approaches address all of these requirements. An important set of approaches [47, 55, 57, 62] train various machine learning approaches to predict the privacy labels of images. The size of the data they need changes based on the trained model but many of them require a large amount of data to train accurate classifiers because of the model complexity. To satisfy the need for big data, these approaches use the data as a single training set by assuming the data come from a single source, such as a dataset available in an OSN. Thus, the personal data requirement would be omitted. This typical training process results in a single classifier that is the same for all users, and therefore, the predicted privacy settings would not comply with the privacy variance requirement.

On the other hand, there have been approaches [30, 48, 64] suggesting solutions to the privacy variance requirement. These models also face the cold start problem: when

there are small data to learn from, these approaches cannot make accurate predictions. This typically takes place when a user is new to the system or has not shared much. In order to solve the cold start problem, they propose various methods such as building a different classifier for each group of users with similar privacy preferences [64], finding a privacy policy from another user in the network [48], or asking others that the users trust for privacy policy recommendations [30]. Even though these approaches satisfy the privacy variance requirement at some extent, they rely on non-personal data to make predictions, violating the personal data requirement (see Sect. 5 for a more detailed comparison).

To accommodate the above requirements, we take an agent-based approach. Our proposed approach represents each user with a software agent, which helps its user set the privacy of her contents by recommending privacy settings. To respect the personal data requirement, the agent will only access the contents its user shares or the contents shared with its user. Since the privacy expectations vary for each user, each agent will learn the privacy expectation of its user, rather than a general privacy understanding of a system; thus supporting the privacy variance requirement. When an agent is learning its user's privacy expectations through images, one obvious choice is to employ classical machine learning techniques on images. However, it is well-known that these techniques require large data sets for training, which we aim to avoid in this work as it might not be readily available.

We develop a novel privacy model to learn and recommend privacy settings, which is inspired from ideas in information retrieval. The privacy model represents contents using their associated *tags*. Existing works show that tags of an image are successful indicators of content. When images are the subjects, automated systems can use tags to define access-control policies [31]. The tags of an image can be set by the user herself as well as generated automatically by tools. This makes it possible to decide the privacy of an image based on given tags. Moreover, in a recent study, Fogues et al. [19] analyze how tags and tie strength jointly are employed to specify access control policies for photo sharing. Their results show that tags and tie strength are extensively employed by users to define a privacy policy instead of using default privacy policies as is done in Facebook. Our proposed model is generic and can be realized differently with different agent designs.

We present an agent, PELTE that uses the proposed privacy model for recommendation. In PELTE, 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 setting for new images using the privacy retrieval model. Important generic aspects of the model, such as dealing with unknown tags and mimicking others when the user has not shared enough, are made concrete. PELTE does not require any predefined set of tags or any specific input space representation to be in place. Thus, it can be used within any system where the content can be represented using tags; thus respecting our requirement of robustness. Our evaluation shows that using only the tags of images, PELTE is able to predict privacy setting of contents accurately, even when the number of tags or the number of shared contents is low.

The rest of the paper is organized as follows. Section 2 develops our formal framework and describes our model for privacy retrieval. Section 3 presents our agent PELTE that realizes the proposed privacy model, with details on its implementation. Section 4 evaluates PELTE in various experiments over multiagent simulations and presents the results. Finally, Sect. 5 discusses PELTE in relation to the existing works in the literature and illustrates some of the future directions.

2 Privacy retrieval model

Each user of the OSN is only connected to a certain other subset of users with some predefined relations, such as friendship, and share various type of content with them. Users cannot see every shared item on the OSN, rather they are only allowed to see content that are shared with them per se. We propose each user to be supported by a software agent to manage the privacy settings of their posts. The agent acts to help the user and can view all the posts that are available to the user; i.e., the posts that are shared with the user as well as the ones user shares. Posts may contain various content types, such as text, image, or video. We specifically focus on one content type, image, which is in a great demand. It is common practice that people add tags to their images to make their images more visible and understandable for other users who see and search for them. A tag is a keyword such as "woman" or "beach" that either identifies an object in the image or reflects the context of the post. These tags might have been produced by the users as well as an automated tool, such as Clarifai [10]. Whenever a content is assigned a privacy setting to be put online, we consider it as a post.

Definition 1 (*Content*) A content is a tuple $c = \langle i, T \rangle$, where i is the image in the content and T is the set of tags associated with the image. T_c to refer to the tags of a given content c. Note that by using different tags, the same image can be made into a different content. This is intentional and is useful to demonstrate the effects of the choice of tags for a content.

Definition 2 (*Post*) A post is a tuple $p = \langle c, S \rangle$, where c is the content in the post and S is the privacy setting of the post. The content and the privacy setting are visible to users who can view the post. We use S_p and c_p to refer to the privacy setting and the content of the post p and T_p to refer to the tags of the post, such that $T_p = T_{c_p}$.

An OSN user can build a network consisting of connections to her friends as well as her acquaintances from various relation types, such as colleague. The user needs to organize her network with respect to her experience on the OSN, as she does in the real life. Misra and Such [43] analyze the top ranked social media sites and classify them according to the control mechanisms they provide. The authors find that although most of the OSNs that we use today support only one type of relationship—usually called as friends—they provide some additional features to allow users distinguish their friends. For instance, Facebook users can connect to their network with only friend relationship but they can also categorize their friends by creating computer-supported lists. Similarly, LinkedIn users can organize their network by using the predefined groups. On the other hand, OSNs may also support different types of relationship and present those with different names, such as *Friend*, *Colleague*, and *Family*, that match up to the use cases of OSN more [20]. Definition 3 captures this facility of OSNs.

Definition 3 (*Relationship*) $R = \{1, \ldots, m\}$ is the set of all relationship types, which are possible to occur in the system. Users and thus their agents can be connected to each other through these types to yield a relationship. That is, each relationship is a unidirectional connection, denoted with 3-tuple $\langle a, r, b \rangle$, where r is a relation type, such as friend or colleague, from agent a to agent b.

Users upload images of various contents to their OSN accounts but they do not share all images with everyone in their network. In principle, a privacy setting that is used to specify whom the post should be shared can contain various audience groups, such as sets of users, but here we consider the privacy settings in the type of Relationship Based Access Control (ReBAC) that considers the relationship type between users to regulate accesses [21]. ReBAC enables OSN users to specialize their connections in the network by these poly-relational means, resulting in a more natural way to share personal information [20, 43]. The privacy setting of a post composes of separate decisions, which are either deny or permit, for each relationship type. If the image has a context that relates to a specific group of audience, then the user prefer sharing the image only with that group of users. For example, an image of a business meeting might be considered as relevant only to users that are colleagues and the user chooses a privacy setting that permits only the users having the relationship type of *Colleague*.

Definition 4 (*Privacy setting*) A privacy setting is a vector, $S = \langle d_1, d_2, \ldots, d_m \rangle$, containing sharing decision, $d_i \in \{0, 1\}$, for each relationship type $r_i \in R$. A sharing decision is either 1 for permitting or 0 for denying the access from the corresponding relationship type. We use S_p to denote the sharing decision of a post p and $S_{p,m}$ to denote the sharing decision of the post p for a relationship type m.

Definition 5 (*Agent*) An agent is a software that represents a user and recommends privacy settings for the posts that the user is considering sharing. The agent can access the posts that the user has shared as well as the ones she can view. We denote the agent as a 3-tuple $a = \langle P, U, M \rangle$, where P denotes the set of posts that are shared by the user of a and U denotes the set of posts that have been shared with the user of a. $P \cup U$ constitutes the posts that a can view. The function $M : C \rightarrow S$ recommends a privacy setting S for a content $c \in C$ that a is considering to share. Based on the result of this function, a can decide to create a post p with the given content c. We refer to agent a's posts as P_a, viewed posts as U_a, and the recommendation function as M_a.

For any image that is visible to a user, we assume that the user can view the privacy settings and the tags of the image; thus, the agent can obtain the privacy settings and the tags of the images in U_a as is common in many OSNs such as Facebook, where an icon indicating the privacy settings of the images to let users know which other users can see their likes and comments on that image. Whenever a user is interested in sharing a new post p, its privacy setting S needs to be configured. In current OSNs, users are expected to done this by themselves or to use the default settings. However, managing the settings can be complex and the default settings do not satisfy user preferences [60]. Here, the ultimate goal of each agent a is to recommend a setting to its user for each image, by using their function M_a. The recommended setting could be seen as the default setting provided by the system and thus, the user would not be aware of the agent. Such an agent does not introduce any new biases to user behavior or changes in user experience while assisting the user in picking the right options [5]. For each recommended privacy setting, the user is free to follow it or to override it as she sees fit.

Example 1 A user shares images in Fig. 1 in an OSN, where family, friend, and colleague are the relationship types, $R = \{1, 2, 3\}$ respectively. The image in Fig. 1a is taken at a

(a) people, many, festival, crowd, group, exhibition, school, child, carnival, education, ceremony, class, meeting

(b) modern, steel, architecture, futuristic, urban, window, ceiling, construction, building, sky, city, office, light

Fig. 1 Example images and their tags that are generated by Clarifai

meeting in her child's school and therefore, the user targets her family and friends as the audience of the image, i.e. $S = \langle 1, 1, 0 \rangle$. For the image in Fig. 1b, she considers it as an art photo of a modern building and shares with everyone in her network, i.e. $S = \langle 1, 1, 1 \rangle$.

We limit the available data to be used by agents to the content of images, which is in the form of tags, as opposed to metadata of images or other personal information of users. The study of Klemperer et al. [31] shows that the tags of images are successful enough to estimate privacy setting of images. Accordingly, automated systems can benefit from tags to define privacy policies when images are the subjects. Patterns can be found in the tags of images that a user permits a relationship type if the user has consistent decisions about what to share with the relationship type. However, if the same tag appears in different images that permit and deny the same relationship type, then it is difficult to reveal the relation between the tag and the privacy decision even if there is any. In other words, the patterns are expected to be more apparent if the user's previous decisions about the shared 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, agents can use the patterns to recommend privacy settings for the images users upload. In order to do that, agents need a computational model that reveals the patterns from users' previous tags.

We employ methods that are inspired from information retrieval, in which we can measure the influence of tags on privacy settings. Two significant metrics in information retrieval are *term frequency*, which measures the number of times a term occurs in a document and *inverse document frequency*, which measures whether a term is common in a given corpus. Their multiplication yields how important a term is with respect to a document in a given corpus. This is frequently used in search engines to match search keywords with documents. Our intuition here is to understand which tags are significant in indicating the privacy for an image. For example, if the tag "drink" appears frequently for private images only, then one can conclude that this is a good indicator of privacy. On the other hand, if "person" appears equally in both public and private images, then its strength in indicating privacy is limited. This signals two main differences from information retrieval:

- While information retrieval focuses on the uniqueness of terms in differentiating content, privacy retrieval focuses on consistency.
- While information retrieval can differentiate the strength of terms based on how often they occur in a document, privacy retrieval cannot as each tag occurs only once in each image.

Given the above differences, we devise two metrics to be able to measure how important a tag is in determining the privacy of an image for a given relationship: *image frequency* and *public image frequency*.

Image frequency, *if*(*t*), of a tag *t* measures how many times the tag is seen in the shared posts. It is determined for an agent *a* as follows:

$$if(t) = |\{p \in P_a \mid t \in T_p\}| \tag{1}$$

where P_a is the set of post that the agent *a* has shared and T_p is the set of tags that the content in post *p* has. The higher the *image frequency* of a tag, the more precise information it reveals about the privacy preference on the content.

Public image frequency, *pif*(*t*, *r*), of a tag *t* measures how many times the tag is part of a post that is perceived as public for a relationship type *r*. It is calculated for each relationship type *r*, separately, as follows:

$$pif(t, r) = |\{p \in P_a \mid t \in T_p \ \& \ S_{p,r} == 1\}| \tag{2}$$

The *public image frequency* of a tag denotes how strongly the tag is considered to be public for a relationship type. For any relationship, if the *public image frequency* of a tag is equal to the *image frequency* value, then every content that contains the tag has been shared publicly for the given relationship type. This is rarely the case for most tags. What is more frequent is that, a tag appears in contents that are considered public as well as in other contents that are considered private. To calculate the effect of the tag in determining whether a content is public, we normalize the *public image frequency* of a tag for a relationship type with its *image frequency*. This yields the ratio of contents that permits the relationship type to all contents with the same tag and the result is between 0 and 1. If the value is small, the contents are mostly not shared with the given relationship type. Conversely, the value is close to 1 if the given relationship type is permitted for most of the contents. The values that are around 0.5 show that while many contents 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 having around average value is inconsistent, whereas the user's privacy preference on tags having high and low values is more precise.

This ratio of *public image frequency* of a tag with *image frequency* can be thus used to understand the effect of the tag on determining whether the content is private or not. When an agent is in need of determining the privacy setting of a content, it would consider all the tags and calculate the ratio. However, often a content can come with tags that the agent has not seen before in the user's shared posts. For those tags, it is not possible to calculate the *public image frequency* or the *image frequency*. To address this, we expand on the *public image frequency* and *image frequency* definitions above to define *expected* values to account for the unseen tags. Calculating the ratio on these expected values yields an estimation on the privacy setting of the content.

Expected public image frequency, *v*(*c*, *r*), is calculated based on the tags of the image for a given relationship. It considers the tags of the image in two separate cases: the tags that the agent has seen in the user's shared posts and those that the agent has not seen before. For the former, the agent uses the public image frequency [Eq. (2)]. For the latter, it estimates a default value using a function *dpif*(*t*, *r*), which returns a value between 0 and 1. Depending on the agent, different *dpif*(*t*, *r*) functions can be designed. For example, returning a default value of 0 would mean that the function estimates these tags to be private and a value of 1 would mean the tags are expected to be all public. Another possible realization of this function would be to use the average of all the tags seen so

far; this would provide the idea that new tags are expected to be as public as the previous tags. The *dpif*(t, r) function implicitly covers that users can have varying privacy tendencies for different tags. Equation (3) gives the calculations for the *expected public image frequency*, where T_a is the tags that the agent a has seen before in the user's shared images, i.e., $T_a = \{t \in T_p \mid p \in P_a\}$:

$$v(c, r) = \sum_{t \in T_c} \begin{cases} pif(t, r) & t \in T_a \\ dpif(t, r) & \text{otherwise} \end{cases} \tag{3}$$

Expected image frequency, $w(c)$ is also calculated by considering the tags of the content in two separate cases: those that the agent has seen before (T_a) and the tags that the agent has not seen before. For the former, the expected image frequency is calculated as the image frequency. For the latter, it estimates a default value using a function *dif*(t), which returns a value between 0 and 1. The function resembles *dpif*(t, r) in its usage and can be tailored based on the agent's privacy understanding. For example, the function can yield a default value based on how the agent perceives privacy or what the agent has seen so far. Equation (4) depicts this:

$$w(c) = \sum_{t \in T_c} \begin{cases} if(t) & t \in T_a \\ dif(t) & \text{otherwise} \end{cases} \tag{4}$$

Now, that the *expected public image frequency* and the *expected image frequency* are calculated, it is possible to estimate how likely the content in hand to be public by taking their ratio. The ratio is called *privacy value indicator*, *pvi*(c, r), and it estimates a value between 0 and 1 that reflects the tendency for the content to be shared with a relationship type r. Equation (5) shows the calculation:

$$pvi(c, r) = v(p, r)/w(p) \tag{5}$$

The result obtained from Eq. (5) needs to be converted into a decision of a privacy setting. A naive approach would be to permit access for values above a certain threshold, such as 0.5 and the deny access for values below that. However, this has two drawbacks. First, the calculated value is dependent on the agent and will only have a significance if it is put into the context of previous decisions. For example, consider an agent, for whom all previous posts have yielded an average privacy value indicator of 0.9. If the current content yields a value of 0.7, this would mean that the content is less likely to be public, thought the number if above 0.5. Similarly, for an agent with an average privacy indicator value of 0.1, a content that yields 0.4 might still be considered public, even though the value is below 0.5. Thus, it is also necessary to calculate the average privacy indicator values for the previous posts and interpret the privacy value indicator of the current content accordingly. Second, for cases when the current privacy value indicator is close to the average privacy indicator, making a decision is risky because this signifies that the content can be both private and public. For such cases, we employ *average privacy indicator*, *api*(r), which is a function to convert users' previous privacy decisions for the posts into a value. It is calculated for each relationship r as follows:

$$api(r) = \sum_{t \in T_a} pif(t, r) / \sum_{t \in T_a} if(t) \tag{6}$$

If the privacy value indicator of the image is higher than the average privacy indicator for a given relationship type, the image would be considered more probable to be shared with the given relationship type. However, privacy value indicator that is close to average could easily be misleading. Therefore, we use a threshold θ and require that the privacy value indicator has to be at least θ amount different than the average to estimate a decision. Implicitly, θ incorporates a confidence in the decision making: if the difference between the privacy value indicator and average privacy indicator is less that θ, the decision is uncertain. This uncertainty could come about because the user has shared few contents so far or the current set of tags in question do not indicate a clear privacy decision for the image. In such cases, we use a function, $social(c, r)$, where the agents can benefit from information they have perceived from others.

The intuition of function $social(c, r)$ comes from social learning theory [6], which argues that people observe others in social situations and act like the people they observe. Recent work done in OSNs show that OSN users are affected by other users in the system. For example, in an experimental work of social learning theory in the context of OSNs, Burke et al. [8] show that new members of an OSN closely monitor what their friends are sharing and share similar content. In a different work, Xu et al. [61] show that posts from a user's friends influence the user's posts on Twitter. Accordingly, our model incorporates this by enabling agents to benefit from their neighbors in the OSN by mimicking their sharing behavior when they cannot decide how to share content themselves. Equation (7) gives the estimation function for the privacy setting of a content for each relationship type r, based on Eqs. (5) and (6) as follows:

$$estimate(c, r) = \begin{cases} Permit & \text{if } pvi(c, r) > api(r) + \theta \\ Deny & \text{if } pvi(c, r) < api(r) - \theta \\ social(c, r) & \text{otherwise} \end{cases} \tag{7}$$

Note that in order to put this model in action functions that provide default values, $dpif(t, r)$ in Eq. (3) and $dif(t)$ in Eq. (4) have to defined. Moreover, $social(c, r)$ function has to specified according to the design choice for Social Learning Theory. Using the $estimate(c, r)$ for each possible relationship type creates a valid M function for an agent.

3 PELTE: estimating privacy settings using privacy retrieval model

We present a prototype agent, named PELTE, that realizes the privacy retrieval model with its data structures and procedures. The ultimate goal of PELTE is to assist its owner in managing privacy while sharing images and thus, to make the image sharing process easier. In an OSN, it would be possible that a single user owns an instance of PELTE or many users does. As the needs of the privacy retrieval model are limited to the images, PELTE only accesses to the images shared by and with the user. The agent does not need further information, such as the profile information of the user; hence, it does not access to them. PELTE estimates the privacy setting of an uploaded image and recommends the estimated privacy setting to the user.

Table 1 An example tag table of an agent in an OSN that has three relationship types

Tag name	Image frequency	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

3.1 Tag tables

When a user uploads an image to share, the user agent estimates the privacy setting of the image based on the previous data as explained above. The tags of previous images needs to be stored and processed to compute the required indicators. One option is to keep an inverted index, as mostly done in information retrieval, where the tags can be searched to retrieve the images that they have been seen in. However, this requires recomputation of the metrics unnecessarily. Rather, it is more desirable to store the values of the metrics for the tags that the agent has seen and update the values when necessary. Accordingly, we introduce a data structure, called *tag table*, which is indexed by the names of tags such that each row of the tag table corresponds to a tag t, its image frequency value and its public image frequency values, each as separate columns. This structure is highly efficient in terms of the space complexity since the size of the tag table is proportional to the number of unique tags. Note that the size of the tag table is not fixed and grows as the agent becomes aware of new tags. This dynamic nature of the tag table is a desired outcome in terms of robustness because we assume the set of tags are not known upfront.

Table 1 presents an example tag table of an agent in an OSN that has three relationship types, namely Friend, Colleague, and Family. The first row of the table is the header line representing the names of columns. For each relationship type, public image frequency value is shown separately. The given tag table is just a part of an actual table and sorted

based on the image frequency for the sake of clarity. The top row (e.g., tag "people") shows that the user has shared 95 images with the the tag, where she permits access of her friends only for 12 of these images and her colleagues for 10 of those. However, users who have family relationship with the user have been permitted for 42 of them. In the example, "people" has the highest image frequency since it has been used by the user most frequently, whereas the tags at the bottom of the tag table have been rarely seen in the user's shared images and therefore, those have image frequency value of only 1.

Each agent a collects data of images belong to (P_a) and accessible to (U_a) its own user. These two conceptually different types of data are stored in two separate tag tables: *internal tag table*, which stores the data of images that the user shares herself, and *external tag table*, which stores the data of images that are accessible to the user. The internal tag table is the essential component to make PELTE personalized in respect to the fact that privacy is by nature subjective, mentioned as the principle of privacy variance in Sect. 1. The *external tag table* will be employed in the implementation of *social*(c, r) function.

3.2 Computing indicators

The two tag tables of an agent initially are empty. The agent collects data from the environment over time whenever a new image shared by one of the users in the network. For the images shared by the owner, the agent updates the rows of internal tag table for the corresponding tags according to the privacy setting of the image, as presented in Algorithm 1. If any of the tags is not already stored in the tag table, the agents adds the tag to the tag table (line 3). For each relationship type, in case of permit, the agent increments the public image frequency by one. Otherwise, the value remains the same (line 7). In both cases, the agent increments the image frequency of each tag by one (line 5).

Algorithm 1: Update internal tag table

```
   Input: p, image post
   Data: IT, internal tag table
1  foreach tag t in T_p do
2      if t ∉ IT then
3          IT = IT.add(t)                          // add tag to the internal tag table
4      end
5      IT.if(t) = IT.if(t) + 1                     // update image frequency of the tag
6      foreach r in R do
7          IT.pif(t, r) = IT.pif(t, r) + S_{p,r}   // update pif with the decision
8      end
9  end
```

While the agent of the sharing user is updating its internal tag table, agents of users with whom the image has been shared with, update their external tag tables. In other words, if the user permits the *friend* relationship for the shared image, then all friends of the user see the image and their agents update external tag tables according to the tags and the privacy setting of the image. Note that when agent receives an image, the set of tags belong to the image are attached as a part of the post. In many of the current OSNs, it is a common practice that users share their images with a set of tags to portray the context better. On the other hand, even if the set of tags are not attached to the image, the same process would still be possible through using a built-in facility to generate the tags or requesting tags from an outsource tag generation tool.

Recall that the estimation is done by first calculating the privacy value indicator. This indicator is based on using the public image frequency when the tag is known, but expects

a heuristic to be used when the tag is not known [Eq. (5)]. This heuristic could be based on the tag itself as well as a default value for all the unknown tags. Here we use the average of previous values as the default for the unknown tags. This corresponds to the *average image frequency* for Eq. (4) and to the *average public image frequency* for Eq. (3). Algorithm 2 presents the procedure of the estimation function by using the internal tag table. It first calculates the *average image frequency* of the table (line 1). Then, it searches the internal tag table for each tag 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) to take their *public image frequency* and *image frequency as average values* (line 16) while calculating the *privacy value indicator*. To decide whether the image should be shared with a relationship type, the agent compares the *privacy value indicator* with the *average privacy indicator*. If it is higher than the value, then it shares with the relationship type and adds a permit decision to privacy setting (line 18). Otherwise, it adds sharing action of deny to the privacy setting for the relationship type (Line 20). If the *privacy value indicator* is around the *average privacy indicator* and within the threshold boundaries (Line 21), estimation from internal tag table cannot return a sharing action for the relationship type. Then, it uses the *social*(c, r) function to estimate the decision for the relationship type.

Algorithm 2: Estimate privacy setting

Input: c, content to be estimated
Data: IT, internal tag table, R, relation types
Output: S, estimated privacy setting

1 $aif \leftarrow$ getAverageIF(IT) // get average image frequency of IT
2 $n, x \leftarrow 0$
3 $A \leftarrow$ zeros($A, R.length$) // initalize an array of the size number of relation types
4 **foreach** *tag* t *in* T_c **do**
5 **if** $t \notin IT$ **then**
6 $n = n + 1$
7 **else**
8 $x = x + IT.if(t)$
9 **foreach** *relationship type* r *in* R **do**
10 $A[r] = A[r] + IT.pif(t, r)$
11 **end**
12 **end**
13 **end**
14 **foreach** *relationship type* r *in* $\mathcal{R}$ **do**
15 $apif \leftarrow$ getAveragePIF(IT, r) // get average public image frequency of IT for r
16 $pvi = (A[r] + apif * n)/(x + aif * n)$ // privacy value indicator
17 **if** $pvi > (apif/aif) + \theta$ **then**
18 $S.add(r, 1)$ // permit decision
19 **else if** $pvi < (apif/aif) - \theta$ **then**
20 $S.add(r, 0)$ // deny decision
21 **else**
22 $S.add(r, social(c, r))$ // undecidable state
23 **end**
24 **end**
25 **return** S

3.3 Social estimation

We are inspired from the social learning theory [6] in the sense that users mimic their friends if they do not have certain privacy preferences. This happens especially when a user is a newcomer. From the perspective of a newcomer, an OSN is a union of the previously joined users and the posts that the users have already shared. As the newcomer starts to share her own images, she builds her own privacy preferences over time. Moreover, a user might have not made certain privacy decisions on some context even though she has shared many images. We consult the Social Learning Theory again; but

this time, since the user is not completely inexperienced, she may adapt herself more to some of her friends while ignoring some others. This is, benefiting more from those that have had similar privacy preferences with the user. For example, if two friends share many images with similar tags and the same privacy setting, this would signal that their privacy preferences are similar. Based on this intuition, we analyze the privacy settings of a user's friends' shared images to judge how similar they are to each other.

We use a metric, called *similarity*, to assess how how similar a user's privacy preferences are with a friend on the shared posts. And thus, to benefit more from their privacy decisions privacy preferences. As a result of the ReBAC model, a user's *similarity* to another user yields to a multidimensional value, in which each dimension corresponds to the similarity of privacy preferences at a type of relationship. For each relationship type r, an agent a computes the *similarity* to the user's friend b based on the set of posts that user b has shared with the owner of agent a, i.e., the intersection of U_a, and P_b, as follows:

$$similarity(b, r) = |\{p \in P_b \cap U_a \mid (estimate(c, r) = S_{p,r})\}|/|P_b \cap U_a| \tag{8}$$

This equation finds the images whose privacy setting (the actual decision given by agent b) would be the same with the estimated decision of agent a as if agent a was to actually share the image (using Eq. (7)). Then, it compares the number of them with the total number of images. The more the number of images that users share the same privacy setting, the higher the *similarity* value for both of them or vice versa. If the estimation function resorts to $social(c, r)$, i.e., does not return a privacy decision from the *internal tag table*, then this image would not be considered in the calculation. Since the estimation function does not yield to a certain privacy decision when a user is a newcomer, *similarity* value is assumed to be 1 for each friend until the users starts to have certain privacy preferences. This corresponds to the observation phase of the Social Learning Theory in which they learn how to act from others without judging their actions. Note that the *similarity* values are unidirectional; that is, user a's similarity to user b could be different than user b's similarity to user a. Moreover, *similarity* might be different for each types of relationship.

The agent stores the posts shared with the user, U_a, in the *external tag table* and then uses the table to make decisions with the social estimation function. The data structure of the *external tag table* is the same with the *internal tag table*, except that the *external tag table* contains separate *image frequency* values for each relationship type because the agent uses *similarity* metric as a coefficient in the update procedure and the *similarity* values are different for each relationship type. The difference of the update procedure (Algorithm 1) for the *external tag table* is that the agent uses the *similarity* to the user who shared the image as a coefficient in both the *image frequency* update (line 5) and the *public image frequency* update (line 7). Hence, the posts that are shared by users who have similar privacy preferences have higher impact than the posts that are shared by less similar users.

The procedure of the *social estimation* is presented in Algorithm 3, which is similar to the Algorithm 2. This time the agent uses the *external tag table* of the user instead of the *internal tag table*. Since the *external tag table* has *image frequency* values separately for each relationship type, average *image frequency* value is calculated specific to the given relationship type (line 1). The rest of the algorithm computes the indicator values and finally finds a sharing decision for the given image post. Notice that the algorithm returns a sharing decision only for the given relationship instead of a complete privacy setting.

Algorithm 3: social(c, r)

Input: c, content to be uploaded, r, relationship type
Data: ET, external tag table
Output: d, sharing decision for relationship r

$aif \leftarrow$ getAverageIF(ET, r) // get average image frequency of ET for r
$n, x, y \leftarrow 0$
foreach *tag* t *in* T_c **do**
  if $t \notin T$ **then**
    $n = n + 1$
  else
    $x = x + ET$.if(t, r)
    $y = y + ET$.pif(t, r)
  end
end
$apif \leftarrow$ getAveragePIF(ET, r) // get average public image frequency of ET for r
$pvi = (y + apif * n)/(x + aif * n)$ // privacy value indicator
if $pvi > (apif/aif)$ **then**
  $d = 1$ // permit decision
else
  $d = 0$ // deny decision
end
return d

4 Evaluation

The proposed privacy retrieval model and its realization as PELTE address the four requirements explained in Sect. 1, namely private data, small data, privacy variance and robustness. First, since each agent only sees the posts that it shares and the posts that are shared with it, we satisfy the personal data requirement by design. In other words, agents do not see each others's posts unless they have been shared with them. We analyze if and to what extent, PELTE satisfies the remaining three requirements. To show that PELTE can work with small data, we experiment with variying data availability for each agent (Sect. 4.2). To show that PELTE can accoommodate privacy variance, we experiment with settings where agents are on purpose given contradictory privacy preferences (Sect. 4.3). Finally, to show that PELTE is robust, we experiment with settings where images have few tags or that the images have been labeled differently by different users. These capture cases where the agents have access to missing information or inconsistent information (Sect. 4.4). Overall, we are especially interested in the following questions: Can PELTE predict the right privacy preference, if the user has shared only a few images before? Can PELTE work well if the images have only a few tags? Can PELTE predict correct privacy settings when other agents have contradictory privacy settings?

In order to answer these questions, we make use of multiagent simulations [15]. Multiagent simulations enable a set of agents to execute with predefined system rules over a certain set of time steps. By varying parameters of the simulation and the agents, different simulation setups can be obtained.

4.1 Simulation environment

We have developed a multiagent simulation environment where a set of agents can execute in line with PELTE. The underlying idea of the simulations is to enable agents to create posts to share with others. While doing that, each agent predicts the privacy settings of an image in the post. To do this, the simulations first need a privacy-labeled dataset of images so that

(a) Private Images **(b)** Public Images

Fig. 2 Example images from PicAlert dataset

each agent can make a decision on the privacy of the image and the simulation can check whether this decision was correct against the provided image label. Next, the simulation needs the agents to be connected to each other through an OSN so that each agent can share posts with those they are connected. Finally, the simulation needs a set of rules to describe what will happen at each cycle of the simulation. We explain these three steps in detail next:

Dataset: The images used in the simulation environment are obtained from PicAlert dataset [63]. 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 images that are no longer available on Flickr because we could not generate tags for them. Some of the images have conflicting labels from different evaluators. To avoid the uncertain decisions on the labels, we remove the images with conflicting labels in all experiments except the one that we analyze the effect of these images (Sect. 4.4.3). We select equal number of public images with the private ones, ending up about 7000 images. Examples of private and public images are presented in Fig. 2. For each image, we generate 20 tags by using an automated tool, Clarifai [10], where the tags correspond to concepts, objects, scenes, and so on, as we can see in Fig. 1.

In the dataset, public images have 0.77 unique tags per image, whereas private ones have 0.49 unique tags per image. While average occurrence of a tag is 25.9 for public images, that is 40.8 for the same number of private images. We see that the most frequent tags of private images are more frequent than those of public images. For example, *people* tag has frequency of 0.93 in private images, 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. Another significant 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 [62].

Notice that even though *people* tag is highly dominant for the set of private images, there are still private images that do not have people in it, such as the right image in Fig. 2a. There are also many public images that have people in it, such as the middle image in Fig. 2b. On the other hand, the automated tool cannot be expected to create tags that exactly covers the content of an image. For instance, the image on the right in Fig. 2b has *people* tag despite the main objects in the image are bananas and wooden toys. Therefore, it is not possible that privacy decisions for the images can be easily given via simple decision rules. Moreover, none of the information that we achieved by dataset analysis, such as a predefined set of all possible tags, is provided as an input to PELTE. This is, agents are intentionally ignorant of the characteristics of the dataset and suitable to any kind of image dataset.

Social network: The simulation environment needs a realistic social network structure to be in place. We construct the network by using the Facebook dataset called ego-Facebook[1] obtained from Standford University Network Analysis Project [38]. The dataset has different sized networks. We select a network that allows us to evaluate PELTE's performance with varying number of training sets by using the image dataset. We use the network that contains 59 nodes and 146 bidirectional, friend relationships among the nodes, where each node might have different number of relations. 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.

Simulation cycle: 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 Sect. 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 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.

We have developed the above simulation environment in Java. Each simulation cycle takes a set of images as input and returns the predicted privacy settings of the images. The comparison between the actual privacy labels of the images and the predicted privacy labels results in four groups: *true private* is the set of private images that are predicted correctly as private, *false public* is the set of private images that are predicted as public, *true public* is the set of images that are predicted correctly as public, and *false private* is the set of public images that are predicted as private. The performance of PELTE is evaluated as the overall performance of agents via the following success metrics:

- **Private recall:** the fraction of private images that are successfully predicted as private is called private recall. It reflects to how much a system is successful at preserving users' privacy. Klemperer et al. [31] state that people are more concerned with false allows than false denies while sharing posts in OSNs. Therefore, we present private recall values in each of the results we obtain from simulations to analyze that if PELTE can satisfy the main concern of OSN users about privacy.

$$Private\ Recall = \frac{|True\ Private|}{|True\ Private| + |False\ Public|}$$

[1] http://snap.stanford.edu/data/egonets-Facebook.html.

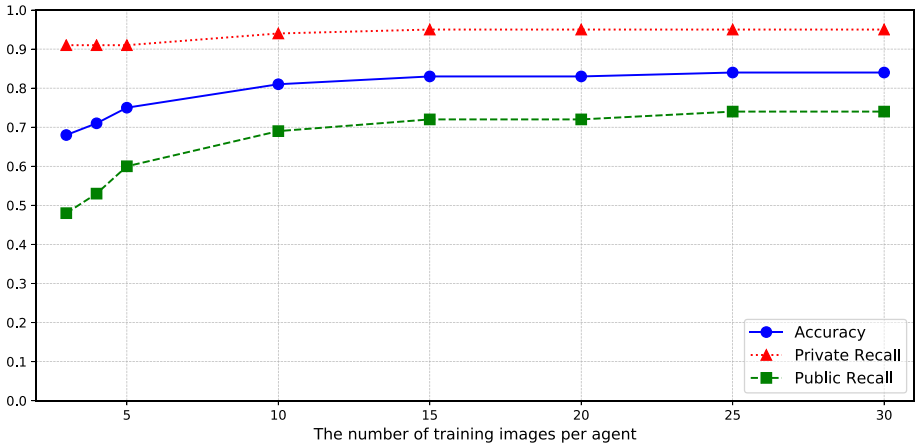


Fig. 3 Accuracy, private recall and public recall values of PELTE when the social function is disabled in the estimation function, i.e., $\theta = 0$

- **Public recall:** the fraction of public images that are successfully predicted as public is called public recall. It is obvious that predicting all images as private would maximize the private recall and preserve users' privacy without any mistake. However, users join OSNs and share their personal information through posting because they intend to share or transmit information to their friends [39]. But their willingness to share their personal data depends on the sensitivity of the data [41]. Therefore, we present public recall values in addition to the private recall values to analyze if PELTE is able to differentiate images that users would share publicly according to both their privacy preferences and the properties of the images. Higher public recall values enable users' shared images to reach other users as much as possible without enforcing unnecessary strictness.

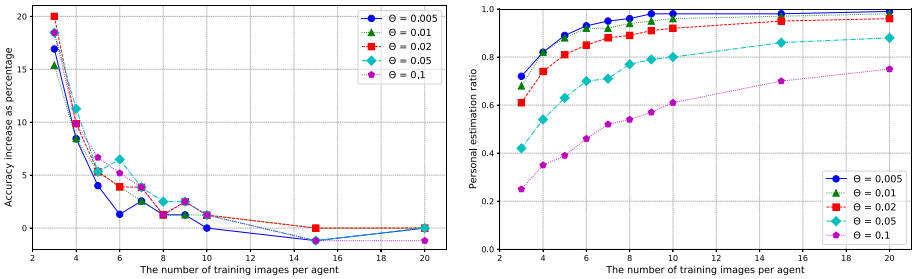
$$Public\ Recall = \frac{|True\ Public|}{|True\ Public| + |False\ Private|}$$

- **Accuracy:** the fraction of both private and public images that are successfully predicted is called accuracy. Private and public recalls measure the success from the perspective of both private and public images. We present accuracy values as the overall success in each of the results to analyze if PELTE would be able to help users manage privacy settings of images.

$$Accuracy = \frac{|True\ Private| + |True\ Public|}{|True\ Private| + |False\ Public| + |True\ Public| + |False\ Private|}$$

4.2 Performance of the estimation function

First, we evaluate the estimation function PELTE when only the internal tag table is available by setting θ to 0. This part mainly focuses on the effect of the number of training images on the accuracy, private recall and public recall values. In each experiment setup, we vary the number of training images and repeat the experiments 20 times. Each agent predicts the



(a) The accuracy increase of the social function is higher with smaller training data

(b) The ratio of personal estimation to all estimations increases with more training data.

Fig. 4 The effect of social function on the estimation function [see Eq. (7)] for varying θ values

privacy setting for 20 test images. Figure 3 presents the average of the results obtained from each experiment setup. The x-axis is the average number of training images per agent in each setup. We then plot the accuracy, private recall, and public recall. The accuracy value of PELTE is around 0.7 when there are 236 training images throughout the system and each agent has seen four images in average. Providing more training data to the agents make the system more accurate, as expected. When each user agent has approximately 25 images, PELTE reaches the accuracy value of 0.85. Moreover, the private recall attains a result that is higher than the accuracy of the system and around 0.95. In other words, PELTE estimates the privacy settings of private images more accurately than those of public images.

These results show that PELTE successfully estimates the privacy of images even when the agents uses only their users' internal tag table. More strikingly, PELTE's success rate in on par with centralized approaches that make use of a far larger data set. In particular, Tonge and Caragea [55] use PicAlert image dataset at one-shot to train an SVM classifier, which achieves an accuracy of 83.14% by using object tags created via ImageNet. In a more recent work [57], they propose an approach for fusing object, scene context, and image tags modalities. The model identifies the set of most competent modalities on the fly and obtains an accuracy of 86.36%. Similarly, Squicciarini et al. [47] use visual features (SIFT, edge direction, facial detection, RGB, sentiment) and tags of images to build a machine learning classifier, which leads to an accuracy of 86.5%.

However, note that estimating only by the means of the internal tag table would cause the agents to face the cold start problem. Because of the lack of training images in the beginning, agents would not be able to learn the users' preferences accurately. For instance, having four images per agent leads to a success of 0.7. The images that have privacy value indicator close to the average privacy indicator are labeled as either public or private even the estimation is not strong enough. Privacy settings of these images are more likely to be incorrect. This is expected to be ameliorated with the contribution of the social estimation function, which uses the external tag table.

We analyze the effect of the social function to the accuracy of the estimation function by varying θ value in Eq. (7) from 0.005 to 0.1. We take $\theta = 0$ as the baseline and then compare the results of the estimation function to the baseline. We expect the results to depend on how much data are stored in the tag tables. Therefore, we observe it under different number of training images. Figure 4a depicts the results, where the x-axis is the number of training images per agent and the y-axis is the improvement at the accuracy given as percentage. Every line represent the results of a different θ value. As clearly seen, the social

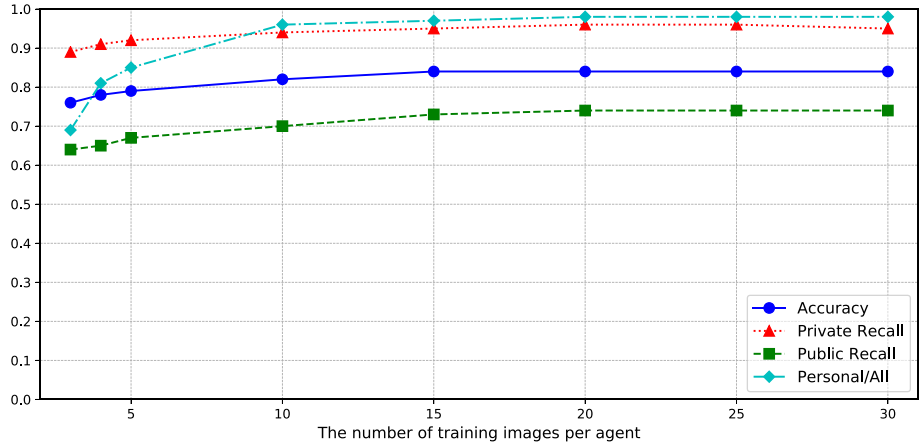


Fig. 5 Accuracy, private recall, and public recall results of PELTE when $\theta = 0.01$

function improves the success rate. The improvement is evident especially when the number of training images is low. For instance, the improvement is around 10% in case of each agent having four training images. If we look at Fig. 3, most of the increase in the results occurs from three to ten training images. This is when the agents have not shared too many posts themselves thus benefit from mimicking the behavior of others by making use of the images that have been shared with them.

On the other hand, increasing θ value does not always make the system much more successful. We analyze how much the social function is involved in the estimation process and present the result in Fig. 4b. The x-axis is again the number of training images per agent and the y-axis is the number of estimations made by using the internal tag table, called as personal, to the overall number of all estimations. Higher θ value causes social function to estimate privacy settings of more images. Moreover, the estimation function needs the social function less with the increasing number of images shared by the agents themselves. However, even though the improvement becomes negligible with the more training images, the social function estimates privacy settings of images for higher θ values, such as $\theta = 0.1$.

Now, we know that using a small threshold value is enough to increase success of the system via social estimation function. We set θ value to 0.01 and analyze the accuracy, private recall, and public recall results of PELTE. Figure 5 presents these results, where the x-axis is the average number of images each agent has. We plot the accuracy, private recall, public recall, and *personal/all*, which shows the ratio of the estimation function only uses the internal tag table to the total number estimations. It helps us to understand how many images are estimated by using the internal tag table. 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 *personal/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 social estimation function on the results more clearly by comparing Figs. 3 and 5. When each agent has only three images and θ is set to 0, the accuracy of the estimation function is less than 70%. However, when the system enables the social estimation function, the accuracy is close to 80%. Moreover, having θ value equals to 0.01 helps the system to reach its maximum success earlier than the system that benefits from only the internal tag table. Even when there are few number of

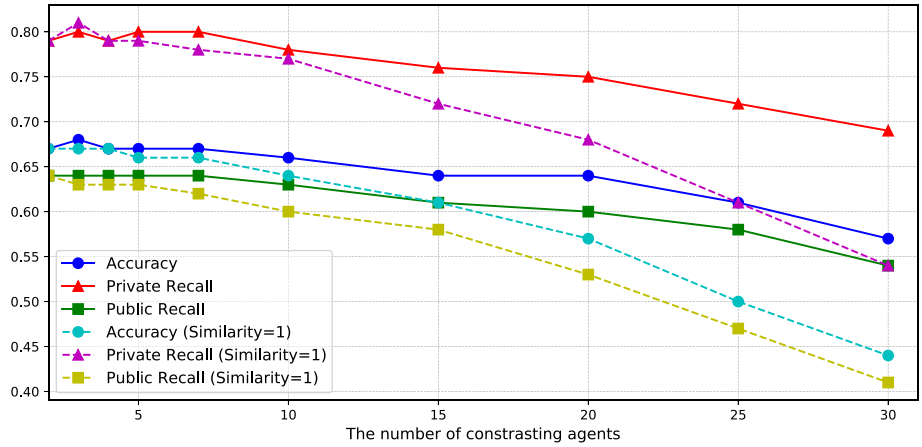


Fig. 6 The results of the social estimation function when the *similarity* metric is in use and not (assumed to be 1)

training images, accuracy and recall values are comparable to the best results that the full system achieves. Note that this is possible because content is being shared with the user even though the user has not shared much herself. Therefore, we can conclude that the social estimation function improves the results of PELTE when it suffers from the cold start problem. If the estimation function only used the internal tag table, it would have required much more data to yield the result that is obtained with the social estimation.

4.3 Performance under privacy variance

In the previous scenarios, each agent is indifferent to the privacy understanding of other agents. The privacy understanding of other agents is not important when an agent is using its *internal tag table*, as this only reflects its own preferences. However, when the agent is using its *external tag table*, the possible privacy variance among agents would become more of an issue. That is, the agent prefers to share an image as private but many of its friends on the network are sharing similar images as public. Accordingly, by making decisions based on what others have shared with the agent might give misleading results. Hence, the agent should only make decisions based on the agents that it has similar privacy understanding. This is represented as the *similarity* metric of PELTE, which is defined in Eq. (8). In order to show how using the *similarity* metric affects agents' privacy setting estimations, we introduce agents with contrasting privacy understanding to the environment. These agents share images with the opposite privacy settings, i.e., sharing a public image as private or vice versa.

We examine the effect of contrasting agents to the whole network by varying the number of contrasting agents. We randomly choose n agents from 59 different agents in the network. In Fig. 6, we plot the accuracy, private recall, and public recall values of the social estimation function both when the *similarity* metric is in use and not. For the latter case, we simply set the *similarity* between all agents to a default value, 1. The x-axis shows the number contrasting agents in the network. These results are for the remaining (normal) agents, i.e. when there are 10 contrasting agents, accuracy values correspond to the average accuracy of remaining 49 agents in the environment. We see that as the number

of contrasting agents increases, the accuracy and the recall values of the social estimation decreases considerably when *similarity* metric is not actively used by the agents. Even though increasing the number of contrasting agents in the network decreases the success also when the *similarity* metric is in use, the decrease is much slower. The comparison between these two cases shows that *similarity* metric help PELTE selectively learn more from similar agents and decrease the effect of agents with contrasting views. For example, the accuracy of the social estimation function is 30% higher when the half of the network becomes contrasting, i.e., 30 agents.

Notice that the accuracy results presented here are lower than the previous parts because we present the results of only the social estimation. Also, since we want to observe how using the *similarity* metric affects the social estimation function, we increase the θ value from 0.01 to 0.1 to make the function more active, as presented in Fig. 4b. These images are directed to the social estimation because the privacy value indicators estimated from the *internal tag table* are close to the average privacy indicator [Eq. (7)]. Therefore, the estimated privacy settings are more prone to be misclassified.

4.4 Robustness

The simulations use the tags generated by the general model of Clarifai, which provides 20 tags for each image. In all the simulations up to now, we allow agents to use all 20 tags while estimating the privacy setting of an image. In different systems, images might be tagged automatically but with another tool, which generates fewer number of tags than 20 or low quality tags. Instead of using a tool, users might also tag the images themselves and this would result fewer tags and more unique tags. Hence, we experiment the performance of PELTE when images have fewer tags, when tags come from different sources or even when privacy of images are uncertain for users.

4.4.1 Effect of number of tags

To study the effect of number of tags, we first run the simulations where we vary the number of tags used per image and measure the accuracy of PELTE. To investigate the contribution of the social function, we evaluate cases of both θ is equal to 0, i.e., social function is not used, and 0.01, i.e., social function is used. The number of training images per agent is 20 and that of the test images per agent is 20 as well. In Fig. 7, we present the accuracy, private recall, and public values for both of the cases. The x-axis shows the number tags each image has. 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 accuracy is close to 0.7.

On the other hand, we see that having fewer number of tags pushes the system to consult the social estimation function more. This effect can be seen from the line of *personal/all*, which corresponds to the ratio of how many times the internal tag table is used to the total number estimations. Notice that the improvement by the means of social function is negligible when the number of training images is equal to 20 (see Fig. 4a). The social function indeed increases the accuracy in case of few number of tags are present. We can conclude that the social estimation function improves the success of PELTE not only for the cold start phase, but also for the systems having fewer number of tags.

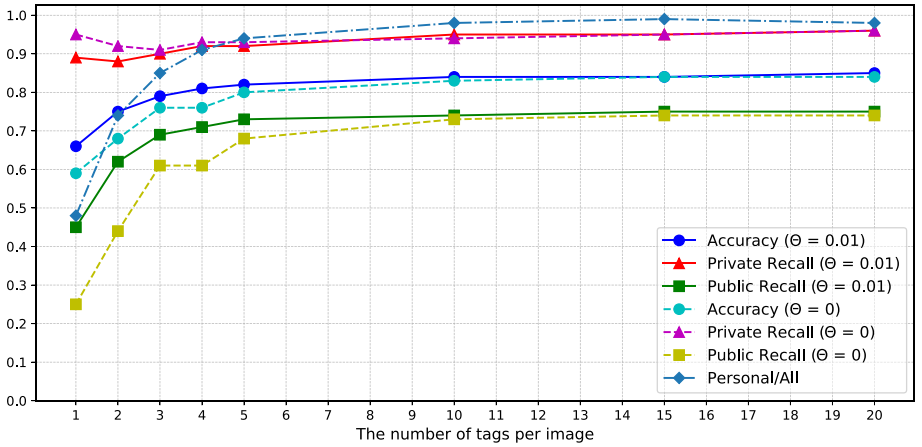


Fig. 7 Accuracy, public recall, and private recall values of PELTE for varying number of tags per image

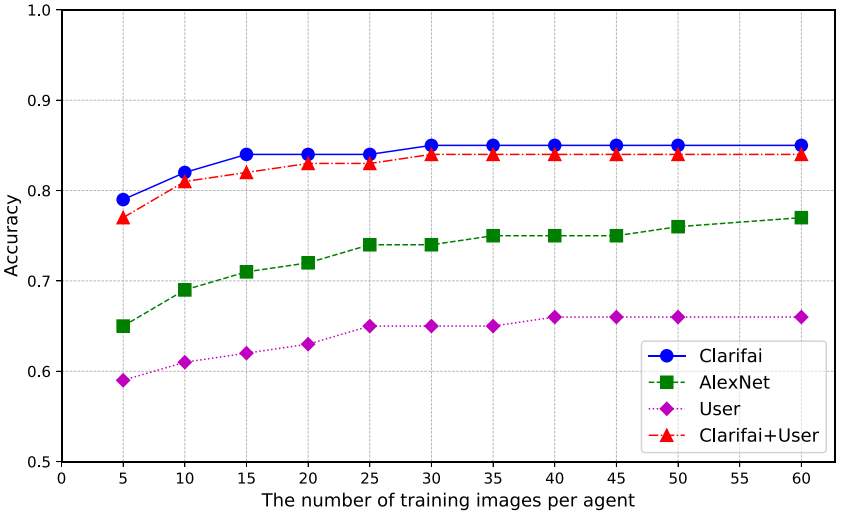


Fig. 8 Accuracy results of PELTE when tags from different datasets are used for the same images

4.4.2 Effect of sources of tags

To see if the quality of the tags affect the performance of PELTE, we run the same experimental setups for the same images: with tags from two different sources: user tags and deep tags[2], which have extracted by Tonge and Caragea [56] using AlexNet convolutional neural network [36]. We evaluate the performance of PELTE by replacing Clarifai tags with the tags of these source for each image. Additionally, we combine the users tags and Clarifai tags of each image and introduce the combinations as the tags of images. We run separate

[2] https://github.com/ashwinitonge/deepprivate

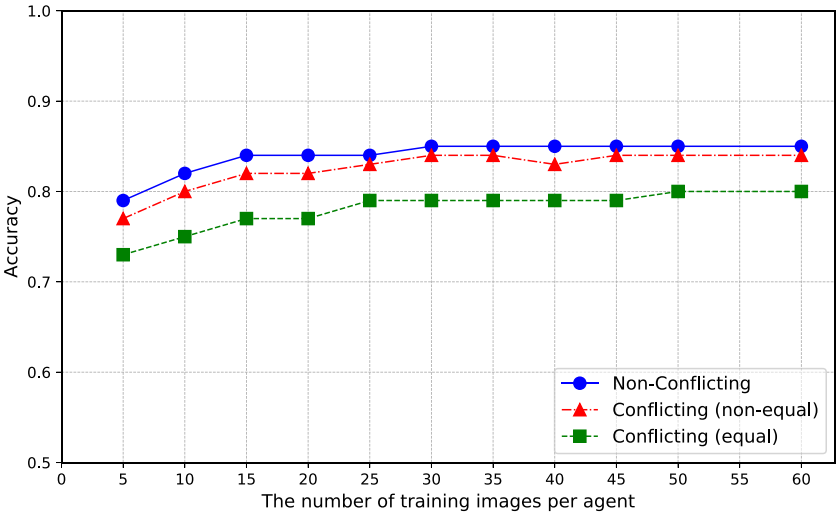


Fig. 9 Accuracy results of PELTE when images having conflicting privacy labels are included

simulations for each of these tag datasets and evaluate the performance for varying number of training images. We present the accuracy results in Fig. 8. The results show that PELTE achieves better results with deep tags than user tags while obtaining the highest accuracy with Clarafia tags. On the other hand, when we use the combination of user and Clarifai tags, PELTE achieves almost the highest performance (0.85).

We have investigated Clarifai, AlexNet, and user tags to find out where the difference stems from. Clarifai and AlexNet generate almost the same number of tags per image (Clarifai generates 20 tags and AlexNet generates about 18.50 tags), whereas the users provide fewer than the half of the number of tags generated by automated tools (8.8 tags). Moreover, the user tags has around unique 2.5 tags per image and this is about five times of the same value for the generated tags. This reveals that automated tools consistently use a smaller set of tags, but users tend to create more unique tags since they tag the images individually. If we consider user tags as noise to the consistent tags of Clarifai, we can conclude that PELTE can overcome the noise and attain almost the same values with its highest level of performance.

4.4.3 Effect of privacy uncertainty

We setup an experiment to see if PELTE performs well when the images cannot be clearly identified as privacy or public. To realize this, we go back to PicAlert dataset and include the images that have conflicting labels; some users label the image as private and some others label as public. We take these images into consideration as two different groups: The first group includes all of the images with conflicting labels except the ones that have the equal number of private and public labels and the second group includes all the images by considering the ones having the same number of private and public labels as private images. The first group has around 20% more images, whereas the second group doubles the number of images. We run simulations with these image groups and compare the performance with the case when we use only images with non-conflicting labels, as in the previous experiments. We present the accuracy results of PELTE for different number of training

images in Fig. 9. The accuracy decreases slightly when images with conflicting but non-equal labels are included. However, when we also include the images that have equal number of private and public labels and consider them as private images, the accuracy value that PELTE achieves at the highest performance decreases from 0.85 to 0.80.

5 Discussion

We propose an agent based approach to assist OSN users manage privacy settings of images. Agents store the tags of uploaded images and then use these tags to automatically recommend a privacy setting for a new image that will be shared. We develop a simulation environment on which we can evaluate the performance of our approach. The environment allows various number of agents to exist and estimate privacy settings at the same time. The tags of the images are obtained from an automated tool. Results, as illustrated in Sect. 4, show that PELTE can estimate privacy setting of images accurately. When each user agent has as few as 25 images, each with 20 tags, PELTE reaches the accuracy value of 0.85. What is more striking is that, the PELTE achieves better performance in predicting that a content is private. This is important because it shows that private content is much less likely to be recommended as public. Repeating the same experiment with as few as five tags yields an accuracy of 0.8; the accuracy drops sharply with fewer than five tags as PELTE has no ground to make recommendations.

An important component of the privacy model is the social function that estimates the setting based on what has been shared with the user. Our first experiment on this aspect is to see when it is good to invoke the social function by varying the θ in Eq. (7). We show that setting θ value as small as 0.01 leads the model to improve accuracy as much as for larger values without invoking the social function excessively. For cases when the social function is invoked, it is most useful when the number of training data is very few. With each additional content that the user shares, the need for the social function drops. After ten contents being shared, the agent does not have to invoke the social function at all. A central question is how much the social function is affected by the privacy variance among other agents. After all, if the agents all have conflicting privacy expectations, mimicking others will not be useful. We observe that as the number of contrasting agents increases, the usefulness of the mimicking drops but still enables higher accuracy than cases without mimicking. Finally, the quality of the tags plays an important role in how PELTE works. When the tags are generated by Clarifai, the accuracy is 0.85, but with user tags the accuracy stays at 0.65. This is an expected result as users might assign tags that are more idiosyncratic than an automated tool. Interestingly, having images with uncertain privacy labels in the dataset decreases the accuracy by at most 0.05. These results are promising in both performance and robustness of PELTE.

5.1 Comparison with State-of-the-art

Recent works in the literature mainly focus on machine learning approaches to predict privacy settings of images more accurately. Squicciarini et al. [47] explore users' uploaded images, using images' visual features (SIFT, edge direction, facial detection, RGB, sentiment) as well as their tags. They employ different machine learning models, such as Naive Bayes, k-nearest neighbors, and support vector machines and evaluate it by using PicAlert dataset. They aim to identify the smallest combination of features that can successfully

lead to highly accurate classification. They find that tags are the most dominant features. Similarly, Tonge and Caragea [55] train SVM classifiers to predict privacy labels of images. The model is a single classifier that is trained on PicAlert images, by using both user tags and deep tags, which are the top 10 object categories identified by a pre-trained ImageNet model. Using top 10 object categories allows to create an input vector space of size 10. Neither of these approaches is applicable for cases with small data. These proposed approaches uses a large set of images to train a single classifier and therefore, do not consider privacy variance requirements. Moreover, the feature vector space needs to be recomputed whenever the training size or the number of selected tags changes. Conversely, PELTE starts from scratch without any assumption about the representation of content and thus satisfies the robustness requirement.

Machine learning has also been used in systems there privacy variance is taken into account. Fang et al. [16] develop a model called Privacy Wizard, which uses active learning methods to help each user set the privacy preferences towards the other users. The privacy wizard requires profile data, such as network connections, age, gender, to assign similarity values to users. It constructs a decision tree that takes a user's labeled friends as inputs to classify the unlabeled ones. However, the proposed model aims to find user's general privacy preferences, whereas PELTE focuses on predicting privacy settings of each image post.

Zhong et al. [64] propose a personalized model to classify images, while acknowledging that the limited user data is too small to train a classifier accurately. Rather than using the tags, their method processes an image into patches to find spatially localized regions and identifies the image as private if there is at least one patch with sensitive content. They consider the approach as a personalized model since they divide a set of users into subsets, which are called privacy groups. Then, the model associates a new user with the group at different strengths based on the user's privacy labels to image patches but also her profile data, which is a 30-dimensional binary vector that corresponds to demographic information. When a new user does not have any labeled image, the system finds her group, based on the profile data. The user profiles here have been used as additional data to help with the small data. The approach does not satisfy the robustness requirement because it needs significant preprocessing and configuration (e.g., number of user groups or vector size) before being used.

Agent-based approaches for privacy prediction also exist. Misra and Such [44] propose an agent based access control decisions by combining content features and social relationships among agents, factoring in type and strength. Each agent is trained with machine learning algorithms, such as SVM and Random Forest over a very large image dataset, where each is represented with a fixed size binary vector of size 15 that corresponds to tag categories. While this approach is decentralized and can learn the preferences per user, the amount of training data used is huge; thus not applicable in systems with small data. Kepez and Yolum [30] also propose an agent-based framework where each agent employs machine learning techniques to learn their users' preferences. To deal with cold start problem, they employ a multiagent approach where an agent asks others that it trusts in the multiagent system for recommendations. They assume that each agent can represent the privacy preferences using a fixed set of features. They show that when the training data is large or that there are trusted agents in the system the agent can help its user. However, this approach does not satisfy the small data requirement as well as the robustness requirement as the input space depends on the training dataset.

Criado and Such [12] propose an Information Assistant Agent that is responsible for managing the interactions of its user in an OSN. The agent uses the information model and

has four main components: community finding algorithm, passing time function, message sending function, and message reception function. It learns the user's behavior in particular contexts and make recommendations for other similar contexts by means of the contextual privacy norms. For example, it warns the user before exchanging a potentially inappropriate information or engaging in an undesirable dissemination of information. Similarly, Ulusoy and Yolum [59] investigate privacy norms specific to the image sharing scenarios. They propose a normative agent-based solution to ease burden on the users in collaborative systems. The agents incorporate four different norm types to provide access control decisions collaboratively. Similar to privacy retrieval model of PELTE, the agents in their approach use tags of images to infer the contextual information. These works on privacy norms are important and complementary to our work. By using the privacy retrieval model proposed here, the agents proposed in these works can estimate the privacy of the content accurately, thereby leading to more accurate privacy norms.

Albertini et al. [3] develop a recommender system that extract association rules from previous contents of a user and combines these rules to generate privacy policies. The proposed model faces with the cold-start problem, whereas social estimation function of PELTE addresses the cold start problem. Similarly, Squicciarini et al. [48] propose a recommender, called Adaptive Policy Prediction (*A3P*), and they consider the cold start problem as well. *A3P* has two components called *A3P-Core* and *A3P-Social*, where *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. However, accessing the entire OSN is both impractical and violates the personal data requirement.

5.2 Connection to other directions in privacy

The idea of helping users manage their privacy through software has been gaining momentum in the past few years. We have proposed an approach specific to privacy settings of images. However, privacy is not only about what users share about themselves but also what others share about them [53]. There are different ways of considering how privacy can be preserved in OSNs. Some of the recent literature analyze information disclosure to determine possible ways of privacy breaches [35, 65] or to predict a user's privacy risk when interacting with other users in OSNs [2]. Several other approaches consider how privacy violations can be detected [33, 49]. 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 [9, 25], argumentation [32], negotiations [29, 54], help of a mediator [52], secret key sharing [28] and so on. We review some of these work in comparison to PELTE here.

Kökciyan and Yolum [33] propose a semantic approach to detect privacy violations in OSNs so that users can take appropriate actions. Three main contributions of the work are meta-model to represent online social networks formally, a semantic model that conforms to the meta-model, and an 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 commitment violation, which corresponds to privacy breach. Their algorithm for detection is both sound and complete, but privacy policies are manually specified by users. PELTE can generate privacy policies automatically and thus can complement the work of Kökciyan and Yolum.

Fogues et. al. [17] propose an agent-based approach, SoSharP, to make effective recommendations about sharing in multiuser scenarios, where a content is about multiple users and thus the users have to decide on the content's privacy together. The proposed approach uses context, user characteristics, sharing preferences, and group characteristics as the relevant features. SoSharP works for three rounds by using different variations of the feature set. In the first round, it starts with context and user based features, whereas it adds sharing preferences in the second and group-based features in the third round. It continues until users agree on the recommendation. After three rounds, if there is no agreement, it is considered a failure. SoSharP is evaluated by conducting a user study in which participants decide for sharing policies of images via impersonation method. Results show that SoSharP has a slightly better performance than veto voting. SoSharP also deals with the cold start problem, but it uses a crowd-sourced training dataset, whereas PELTE does not use any data that are not shared by or with the user.

Humpert et al. [26] survey the interdependent privacy problems and technical solutions in various domains. They have found that almost all the technical solutions focus on either photos or generic data (including photos). Although PELTE has been considered as an agent-based solution for single user scenarios, it would be interesting to employ PELTE as the individual decision making module of such multiuser scenarios. For example, Squicciarini et al. [50] 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. As part of the auction mechanism, PELTE can be used to assign bids to images automatically according to privacy value of the image. Similarly, Such and Rovatsos [54] and Keküllüoğlu *et al.* [29] propose negotiation mechanisms for conflicts in OSNs that support ReBAC. PELTE would be able to act on behalf of users to provide input to the negotiation mechanisms in case user preferences are requested.

OSN users might have difficulties understanding the privacy settings they eventually select for the post they share. Lipford et al. [40] show that providing users visual guidance with a better user interface improves the experience of users. PViz [42] is a graphical tool that display privacy settings at different granularity levels to help users understand whom the privacy settings allow. Such visual designs could be integrated to the implementation of PELTE to explain users the recommended privacy settings and also how they are retrieved from the tags.

5.3 Limitations and future directions

The current work has some limitations and possible areas for further development. We have evaluated the access control mechanism of PELTE for a single relationship type, which is identical to the binary distinction, such as deciding to share with the whole network or only friends. Although the binary distinction is the most widely-used and practical access control mechanism by the current OSNs [43], PELTE can support multiple relationship types, which yields to a more desirable mechanism. An ideal evaluation of PELTE would require a network dataset where the users are connected to each other with multiple relationship types and an image dataset that has privacy labels for the corresponding relationship types. However, the well-known image dataset [63] in the privacy literature is created by impersonation method and cannot be mapped to real users in an OSN network. Therefore, we have experimented the performance of PELTE using datasets that support a single relationship type. Since our approach considers each relationship type independent from

each other, having only a single relationship type does not endanger the validity or the applicability of the approach. If more relationship types were introduced and included in the privacy labels of images, other experiments concerning the differences among relationship types would have been performed as well.

In addition to the relation types, users can benefit from predefined user groups or custom users groups to specify privacy settings at different levels. The predefined user groups are provided by OSNs, whereas the custom ones could be generated by either a facility of the OSNs, such as smart lists in Facebook, or tools designed to help users, such as ReGroup [4] that employs a machine learning approach. PELTE could support predefined groups as the relation types since the groups share the same semantics; but it cannot support groups defined by the users because those groups would be created with different semantics.

Since OSN 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 [13]. As relationships develop and the personal information exchange occurs in OSNs, the role of trust become even more important for the sharing behaviors [58]. However, the real world experience cannot be directly transmitted to a virtual environment as numerical values. Automated tools measure different metrics with respect to the sharing behaviors. For example, BFF [18] predicts tie strength of the relationships of a user with each person in her network. Similarly, the social estimation function of PELTE is implemented as learning from users' networks based on the *similarity* metric and this metric could be considered as the trust of users towards others on privacy preferences. Automated tools could be employed to generate alternatives to the *similarity* metric. Moreover, we currently propose a multidimensional but the same metric for the relationship types. Since the functionality of *similarity* metric is limited to being used as a multiplier in the update function, it is possible to integrate different trust metrics having different meanings for each relationship type. A possible model to incorporate here could be that of FIRE [27], where different types of trust, such as interaction and role-based trust, are employed together.

Online social networks enable users 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 [1]. The system should be able to adapt to these changes immediately. Many existing approaches 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. As a future direction, we want to study how the change in the users' privacy expectation can be handled in PELTE.

Privacy in Internet of Things of is a growing study area of privacy [46]. 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. 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 work to respect people's privacy.

Funding

References

1. Acquisti, A., Brandimarte, L., & Loewenstein, G. (2015). Privacy and human behavior in the age of information. *Science*, *347*(6221), 509–514.
2. Akçora, C., Carminati, B., and Ferrari, E. (2012). Privacy in social networks: How risky is your social graph? In: *IEEE 28th international conference on data engineering* (pp. 9–19). IEEE Computer Society.
3. Albertini, D.A., Carminati, B., & Ferrari, E. (2016). Privacy settings recommender for online social network. In: *IEEE 2nd international conference on collaboration and internet computing* (pp. 514–521).
4. Amershi, S., Fogarty, J., and Weld, D. (2012). Regroup: Interactive machine learning for on-demand group creation in social networks. In: *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 21–30).
5. Baarslag, T., Alan, A. T., Gomer, R., Alam, M., Perera, C., Gerding, E. H., & Schraefel, M. (2017). An automated negotiation agent for permission management. In: *Proceedings of the 16th conference on autonomous agents and multiagent systems* (pp. 380–390). International Foundation for Autonomous Agents and Multiagent Systems.
6. Bandura, A., & Walters, R. H. (1977). *Social learning theory* (Vol. 1). NJ: Prentice-hall Englewood Cliffs.
7. Boyd, D., & Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, *13*(1), 210–230.
8. Burke, M., Marlow, C., and Lento, T. M. (2009). Feed me: Motivating newcomer contribution in social network sites. In: *Proceedings of the 27th international conference on human factors in computing systems* (pp. 945–954). ACM.
9. Carminati, B., & Ferrari, E. (2011). Collaborative access control in on-line social networks. In: *7th international conference on collaborative computing: Networking, applications and worksharing (CollaborateCom)* (pp. 231–240).
10. Clarifai (2020). General model. Retrieved July 13, 2020, from https://www.clarifai.com/model.
11. Colnago, J., Feng, Y., Palanivel, T., Pearman, S., Ung, M., Acquisti, A., Cranor, L. F., & Sadeh, N. M. (2020). Informing the design of a personalized privacy assistant for the internet of things. In: *CHI* (pp. 1–13). ACM.
12. Criado, N., & Such, J. M. (2015). Implicit contextual integrity in online social networks. *Information Sciences*, *325*, 48–69.
13. Cutillo, L. A., Molva, R., & Strufe, T. (2009). Safebook: A privacy-preserving online social network leveraging on real-life trust. *IEEE Communications Magazine*, *47*(12), 94–101.
14. Diaspora*. The diaspora* project. Retrieved July 13, 2020, from https://diasporafoundation.org/.
15. Drogoul, A., & Ferber, J. (1992). Multi-agent simulation as a tool for modeling societies: Application to social differentiation in ant colonies. In: *European workshop on modelling autonomous agents in a multi-agent world* (pp. 2–23). Springer.
16. Fang, L., & LeFevre, K. (2010). Privacy wizards for social networking sites. In: *Proceedings of the 19th international conference on world wide web* (pp. 351–360). ACM.

17. Fogués, R. L., Murukannaiah, P. K., Such, J. M., & Singh, M. P. (2017). Sosharp: Recommending sharing policies in multiuser privacy scenarios. *IEEE Internet Computing*, *21*(6), 28–36.
18. Fogués, R. L., Such, J. M., Espinosa, A., & García-Fornes, A. (2014). BFF: A tool for eliciting tie strength and user communities in social networking services. *Information Systems Frontiers*, *16*(2), 225–237.
19. Fogues, R. L., Such, J. M., Espinosa, A., & Garcia-Fornes, A. (2018). Tie and tag: A study of tie strength and tags for photo sharing. *PLoS One*, *13*, 1–22.
20. Fogués, R. L., Such, J. M., Minguet, A. E., & García-Fornes, A. (2015). Open challenges in relationship-based privacy mechanisms for social network services. *International Journal of Human-Computer Interaction*, *31*(5), 350–370.
21. Fong, P. W. (2011). Relationship-based access control: Protection model and policy language. In: *Proceedings of the 1st ACM conference on data and application security and privacy* (pp. 191–202).
22. Gates, C. (2007). Access control requirements for web 2.0 security and privacy. In: *Proceedings of workshop on web 2.0 security & privacy*.
23. Graham-Harrison, E., & Cadwalladr, C. (2019). Revealed: 50 million facebook profiles harvested for cambridge analytica in major data breach. The Guardian, 2018. Retrieved May 20, 2019.
24. Gürses, S., & Diaz, C. (2013). Two tales of privacy in online social networks. *IEEE Security & Privacy*, *11*(3), 29–37.
25. Hu, H., Ahn, G., & Jorgensen, J. (2013). Multiparty access control for online social networks: Model and mechanisms. *IEEE Transactions on Knowledge and Data Engineering*, *25*(7), 1614–1627.
26. Humbert, M., Trubert, B., & Huguenin, K. (2020). A survey on interdependent privacy. *ACM Computing Surveys (CSUR)*, *52*(6), 122:1–122:40.
27. Huynh, T. D., Jennings, N. R., & Shadbolt, N. R. (2006). An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, *13*(2), 119–154.
28. Ilia, P., Carminati, B., Ferrari, E., Fragopoulou, P., & Ioannidis, S. (2017). SAMPAC: Socially-aware collaborative multi-party access control. In: *Proceedings of the 7th ACM conference on data and application security and privacy* (pp. 71–82). ACM.
29. Keküllüoglu, D., Kökciyan, N., & Yolum, P. (2018). Preserving privacy as social responsibility in online social networks. *ACM Transactions on Internet Technology*, *18*(4), 42:1–42:22.
30. Kepez, B., & Yolum, P. (2016). Learning privacy rules cooperatively in online social networks. In: *Proceedings of the 1st international workshop on AI for privacy and security* (pp. 3:1–3:4). ACM.
31. Klemperer, P., Liang, Y., Mazurek, M., Sleeper, M., Ur, B., Bauer, L., Cranor, L. F., Gupta, N., & Reiter, M. (2012). Tag, you can see it! using tags for access control in photo sharing. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 377–386). Association for Computing Machinery.
32. Kökciyan, N., Yağlıkçı, N., & Yolum, P. (2017). An argumentation approach for resolving privacy disputes in online social networks. *ACM Transactions on Internet Technology*, *17*(3), 27:1–27:22.
33. Kökciyan, N., & Yolum, P. (2016). Priguard: A semantic approach to detect privacy violations in online social networks. *IEEE Transactions on Knowledge and Data Engineering*, *28*(10), 2724–2737.
34. Krasnova, H., Spiekermann, S., Koroleva, K., & Hildebrand, T. (2010). Online social networks: Why we disclose. *Journal of Information Technology*, *25*(2), 109–125.
35. Krishnamurthy, B., & Wills, C. E. (2009). On the leakage of personally identifiable information via online social networks. In: *Proceedings of the 2nd ACM workshop on online social networks* (pp. 7–12). ACM.
36. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems* (pp. 1097–1105).
37. Lampinen, A., Lehtinen, V., Lehmuskallio, A., & Tamminen, S. (2011). We're in it together: Interpersonal management of disclosure in social network services. In: *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 3217–3226).
38. Leskovec, J., & Mcauley, J. J. (2012). Learning to discover social circles in ego networks. In: *Advances in neural information processing systems* (pp. 539–547).
39. Li, K., Lin, Z., & Wang, X. (2015). An empirical analysis of users' privacy disclosure behaviors on social network sites. *Information & Management*, *52*(7), 882–891.
40. Lipford, H. R., Besmer, A., & Watson, J. (2008). Understanding privacy settings in facebook with an audience view. In: *Proceedings of the 1st conference on usability, psychology, and security*. USENIX Association.
41. Malhotra, N. K., Kim, S. S., & Agarwal, J. (2004). Internet users' information privacy concerns (IUIPC): the construct, the scale, and a causal model. *Information Systems Research*, *15*(4), 336–355.

42. Mazzia, A., LeFevre, K., & Adar, E. (2012). The pviz comprehension tool for social network privacy settings. In: *Proceedings of the 8th symposium on usable privacy and security*.
43. Misra, G., & Such, J. M. (2016). How socially aware are social media privacy controls? *IEEE Computer*, *49*(3), 96–99.
44. Misra, G., & Such, J. M. (2017). PACMAN: Personal agent for access control in social media. *IEEE Internet Computing*, *21*(6), 18–26.
45. Sadeh, N. M., Hong, J. I., Cranor, L. F., Fette, I., Kelley, P. G., Prabaker, M. K., et al. (2009). Understanding and capturing people's privacy policies in a mobile social networking application. *Personal and Ubiquitous Computing*, *13*(6), 401–412.
46. Samani, A., Ghenniwa, H. H., & Wahaishi, A. (2015). Privacy in internet of things: A model and protection framework. In: *Proceedings of the 6th international conference on ambient systems, networks and technologies* (Vol. 52, pp. 606–613). Elsevier.
47. Squicciarini, A. C., Caragea, C., & Balakavi, R. (2017). Toward automated online photo privacy. *ACM Transactions on the Web*, *11*(1), 2:1–2:29.
48. Squicciarini, A. C., Lin, D., Sundareswaran, S., & Wede, J. (2015). Privacy policy inference of user-uploaded images on content sharing sites. *IEEE Transactions on Knowledge and Data Engineering*, *27*(1), 193–206.
49. Squicciarini, A. C., Paci, F., & Sundareswaran, S. (2014). Prima: A comprehensive approach to privacy protection in social network sites. *Annals of Telecommunications-Annales des Télécommunications*, *69*(1–2), 21–36.
50. Squicciarini, A. C., Shehab, M., & Paci, F. (2009). Collective privacy management in social networks. In: *Proceedings of the 18th international conference on world wide web* (pp. 521–530). ACM.
51. Strater, K., & Richter, H. (2007). Examining privacy and disclosure in a social networking community. In: *Proceedings of the 3rd symposium on usable privacy and security* (Vol. 229, pp. 157–158). ACM.
52. Such, J. M., & Criado, N. (2016). Resolving multi-party privacy conflicts in social media. *IEEE Transactions on Knowledge and Data Engineering*, *28*(7), 1851–1863.
53. Such, J. M., & Criado, N. (2018). Multiparty privacy in social media. *Communications of the ACM*, *61*(8), 74–81.
54. Such, J. M., & Rovatsos, M. (2016). Privacy policy negotiation in social media. *ACM Transactions on Autonomous and Adaptive Systems*, *11*(1), 4:1–4:29.
55. Tonge, A., & Caragea, C. (2016). Image privacy prediction using deep features. In: *Proceedings of the 13th AAAI conference on artificial intelligence* (pp. 4266–4267). AAAI Press.
56. Tonge, A., & Caragea, C. (2018). On the use of "deep" features for online image sharing. In: *Companion proceedings of the the web conference* (pp. 1317–1321). International World Wide Web Conferences Steering Committee.
57. Tonge, A., & Caragea, C. (2019). Dynamic deep multi-modal fusion for image privacy prediction. In: *The World Wide Web conference* (pp. 1829–1840). Association for Computing Machinery.
58. Tsay-Vogel, M., Shanahan, J., & Signorielli, N. (2018). Social media cultivating perceptions of privacy: A 5-year analysis of privacy attitudes and self-disclosure behaviors among facebook users. *New Media & Society*, *20*(1), 141–161.
59. Ulusoy, O., & Yolum, P. (2020). Norm-based access control. In: *Proceedings of the 25th ACM symposium on access control models and technologies* (pp. 35–46).
60. Watson, J., Lipford, H. R., & Besmer, A. (2015). Mapping user preference to privacy default settings. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *22*(6), 32:1–32:20.
61. Xu, Z., Zhang, Y., Wu, Y., & Yang, Q. (2012). Modeling user posting behavior on social media. In: *SIGIR* (pp. 545–554). ACM.
62. Yu, J., Zhang, B., Kuang, Z., Lin, D., & Fan, J. (2017). iprivacy: Image privacy protection by identifying sensitive objects via deep multi-task learning. *IEEE Transactions on Information Forensics and Security*, *12*(5), 1005–1016.
63. Zerr, S., Siersdorfer, S., Hare, J., & Demidova, E. (2012). Privacy-aware image classification and search. In: *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 35–44).
64. Zhong, H., Squicciarini, A., Miller, D., & Caragea, C. (2017). A group-based personalized model for image privacy classification and labeling. In: *Proceedings of the 26th international joint conference on artificial intelligence* (pp. 3952–3958).
65. Zhou, M. X., Nichols, J., Dignan, T., Lohr, S., Golbeck, J., & Pennebaker, J. W. (2014). Opportunities and risks of discovering personality traits from social media. In: *Extended abstracts on human factors in computing systems* (pp. 1081–1086). Association for Computing Machinery.