# Learning to Share: Engineering Adaptive Decision-Support for Online Social Networks

Yasmin Rafiq*, Luke Dickens†, Alessandra Russo*, Arosha K. Bandara‡, Mu Yang‖, Avelie Stuart §,
Mark Levine §, Gul Calikli**, Blaine A. Price‡, and Bashar Nuseibeh‡¶
*Imperial College London, UK: {y.rafiq, a.russo}@imperial.ac.uk
†University College London, UK: {l.dickens}@ucl.ac.uk
‡The Open University, UK: {a.k.bandara, b.a.price, b.nuseibeh}@open.ac.uk
§University of Exeter, UK: {a.stuart, m.levine}@exeter.ac.uk
¶Lero, University of Limerick, Ireland, UK: {bashar.nuseibeh}@lero.ie
‖University of Southampton, UK: {Mu.Yang}@soton.ac.uk
**Chalmers & University of Gothenburg, Sweden: {gul.calikli}@gu.se

***Abstract*—Some online social networks (OSNs) allow users to define *friendship-groups* as reusable shortcuts for sharing information with multiple contacts. Posting exclusively to a friendship-group gives some privacy control, while supporting communication with (and within) this group. However, recipients of such posts may want to reuse content for their own social advantage, and can bypass existing controls by copy-pasting into a new post; this *cross-posting* poses privacy risks.**

**This paper presents a *learning to share* approach that enables the incorporation of more nuanced privacy controls into OSNs. Specifically, we propose a reusable, adaptive software architecture that uses rigorous runtime analysis to help OSN users to make informed decisions about suitable audiences for their posts. This is achieved by supporting dynamic formation of recipient-groups that benefit social interactions while reducing privacy risks. We exemplify the use of our approach in the context of Facebook.**

## I. Introduction

Online Social networks (OSNs) are increasingly used to maintain social ties with family members, friends, and colleagues, and build new social relationships (e.g., [1]–[3]). However, these benefits come with an increased risk of *privacy violation* from oversharing or underusing privacy controls [4]–[8]. Many OSN platforms currently support privacy management through features such as static (user-defined) *friendship groups* as reusable shortcuts for sharing a single post with multiple contacts. Users can select the group they deem most appropriate when posting a message. In OSNs such as Facebook, LinkedIn and Google+, users may perceive posting in closed group as a *quick fix* to their privacy concerns, since these OSNs can constrain the re-sharing to only those contacts who have received the original message. However, these privacy control mechanisms do not account for *cross-posting*, i.e. when a contact copy-and-pastes the original message into a new post, and sends it to contacts outside the original group, to either improve their own social capital or damage that of the original user [9]. We argue that privacy in OSNs cannot be effectively delivered by inflexible and rarely-visited privacy settings, instead contact lists should be formed *dynamically* per post, such that unwanted cross-posting is minimized while the user's social benefit is optimised.

In this paper, we propose an adaptive privacy control approach, called *learning to share*, that enables software engineers/developers to incorporate adaptive privacy decision support into OSN applications. Specifically, we propose a software architecture that supports continuous monitoring of online interactions between each user-contact pair in a user's social network, to predict three categories of contacts: those who are most likely to pose a privacy breach (i.e. risky friends); those who are socially inactive but privacy aware (i.e., safe friends); and those who are both socially active and privacy aware (i.e., super friends). The prediction is based on an interaction model of sharing behaviours, which is updated on-line in response to monitored behaviour and used to evaluate social benefit and privacy risk of sharing a post with each potential recipient. The outcome of the classification is used by the decision support component of the architecture to dynamically form contacts lists per post, and allows the user to efficiently select the recipient group on a per post basis. The underlying OSN infrastructure can take into account the informed selection by the user and dynamically control who should be receiving which message when delivering the post.

Our proposed software architecture is not specific to a particular OSN environment. Software engineers working on developing and/or improving existing OSN applications with adaptive privacy decision mechanisms, can use our proposed architecture by deploying its automated learning and decision support capabilities, and integrating an Abstract Interaction Model and a wrapper for monitoring the type of social interactions that are specific to the particular OSN application.

The rest of the paper is organized as follows. Section II presents a motivating example within the context of Facebook. Section III describes our *Learning to Share* architecture. Section IV exemplifies how OSN behaviours can be modelled as parametric Markov model, using the example of Facebook; introduces the related component for monitoring social network activities, and presents the online learning algorithm used to predict the model's parameters. Section V describes the computation of *social benefit* and *privacy risk* used by the decision support component. Section VI discusses related work and concludes the paper with a summary and future work.

## II. Motivating Example

Here, we illustrate the need for adaptive privacy control with a motivating example based on Facebook use. Facebook user *Bob* shares sensitive post A with his predefined *close_friends* group, which does not include *Ann* – Bob's work colleague but not a close friend. As Ann is not a member of the *close_friends* group, Facebook does not allow her to see Bob's post, and members of *close_friends* are unable to *re-share* the post with her. *Tom*, a member of Bob's *close_friends* group, is also a close friend of Ann, but is unaware of this privacy aspect of Facebook's group sharing. When Tom receives the post A, he naively decides to copy and paste the content into a new post, B, adding some content of his own, and shares B with a group of contacts which includes Ann. Now Ann can see, via *cross-posting*, what *Bob* shared in post A, but Facebook is unable to detect this or notify *Bob* that his on-line close friend *Tom* has violated his privacy, albeit unintentionally.

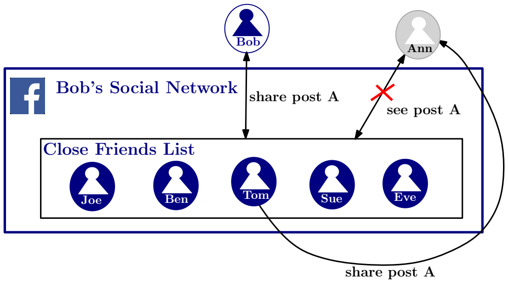

Fig. 1: A scenario of privacy breach.

This shows that, while customised social network groups are convenient for posting to multiple recipients and (in Facebook at least) offering a form of privacy control, this static feature can easily be bypassed, even fairly innocently. Good adaptive privacy control would, instead, monitor Tom's actions, detect the similarity between posts A and B, and use this information to learn that sending sensitive posts to Tom represents a privacy risk, presenting this information to Bob the next time he intends to share another sensitive post with close friends.

## III. Privacy Aware Sharing Architecture

Our proposed *learning to share* approach is implemented as a reusable, adaptive software architecture as shown in Figure 2. The architecture comprises two main modules: i) modelling and monitoring (marked C2), and ii) adaptive decision support (marked C3). This design enables software engineers working on OSN applications to deploy our architecture by reusing the automated learning and decision support capabilities and developing just two components: the abstract interaction model for the specific OSN application and a wrapper, which enables monitoring of the application specific social network interactions, and execution of sharing decisions.

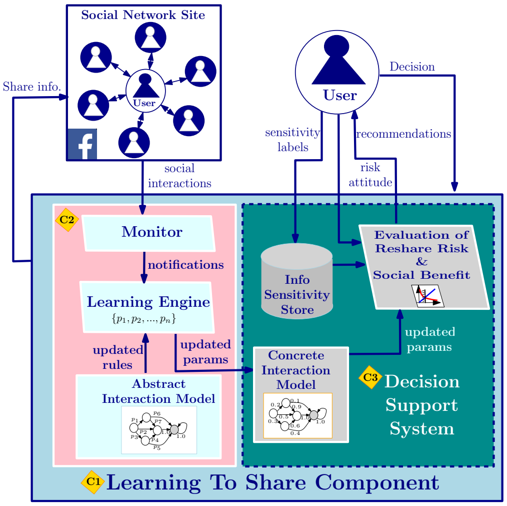

Fig. 2: *Learning to share* Architecture

The OSN wrapper implements the monitor functionality to detect for each recipient of each post subsequent interactions between user and recipient, e.g., resulting *likes* and *comments* on Facebook. Monitored events are used by the *Learning Engine* component, which updates the parameters of the associated instantiated interaction model, called *Concrete Interaction Models*, essentially capturing the probability of each possible social interaction at each post's sensitivity level. The concrete interaction model is then queried by the *Decision Support System* (DSS) to assess sharing decisions. Specifically, the DSS employs the concrete interaction model, a risk averseness threshold and a post's sensitivity level to evaluate the re-share risk and social benefit and to classify (at run-time) each of user's contacts as *super*, *safe* or *risky*. Sensitivity levels of shared posts are defined by the user and stored, together with the post, to check for future re-shares of the post.

By providing just the wrapper and an abstract interaction model, *learning to share* can be deployed for different OSNs either as a fully integrated feature of an OSN's infrastructure (i.e. total view mode), or as a plugin application of an existing OSN infrastructure (i.e. partial view mode). In the first case, the adaptive privacy control benefits from full access to the OSN's global network community and the monitoring of online social behaviour of all members in the network. Whereas, partial view mode only provides adaptive privacy control and monitoring of users of the plugin application.

## IV. Modelling and Monitoring Runtime Activity

This section describes the modelling module of our architecture as a parametric Markov model, illustrating it with an example of an abstract interaction model for Facebook. It also presents our online learning method for updating the model's parameters and a method for monitoring social interaction events, used for computing the updates.

### A. Quantitative Verification of Markov Chains

Online social interactions can be modelled using parametric Markov chains (PMC). These are defined as follows:

*Definition 4.1:* A reward-annotated finite state discrete-time *parametric Markov chain* (PMC), $\mathcal{M}$, is the tuple

$< S, s_0, \mathcal{V}, \mathbf{P}, \iota, L >$, with $S$ a finite set of states and initial state $s_0 \in S$; $\mathcal{V}$ a set of real-valued parameters; $\mathbf{P}$ a parametric transition probability $|S| \times |S|$-matrix whose elements are functions of $\mathcal{V}$; the *reward function* $\iota : S \rightarrow \mathbb{R}_{\geq 0}$ assigning a non-negative reward for each state, and *labelling function* $L : S \rightarrow 2^{AP}$ assigning a set of atomic propositions to each state. The $(i, j)$th element of $\mathbf{P}$, $p_{ij}$, is a function of parameters $\mathcal{V}$ (written $p_{ij} \in \mathcal{F}_{\mathcal{V}}$) and represents the probability of transitioning from $s_i$ to $s_j$. $p_{ij}$ takes values strictly in $[0, 1]$ and $\sum_j p_{ij} = 1$ for all $i$.

Probabilistic model checker PRISM[1] [12] allows PMCs to be expressed in a high-level language, and efficiently evaluates queries expressed in a reward-augmented version of probabilistic computational tree logic (PCTL) [13]. We use PRISM's reward query operator $\mathbf{R}_{=?}[\Phi]$ in conjunction with the reachability reward property $\Phi = [\mathbf{F}\ a]$, to evalute the average reward accumulated along a path until a state satisfying proposition $a \in AP$ is reached (for PCTL semantics see [14]). When queried with a PMC, PRISM's reward query operator produces a symbolic expression, $V \in \mathcal{F}_{\mathcal{V}}$, of the associated reward property (a function of $\mathcal{V}$). In what follows we describe how we use this functionality to predict expected social benefit and privacy risk in the context of Facebook.

### B. Modelling Sharing Behaviours

We describe here our PMC model for Facebook. It captures the online social interactions between a user and contact, $c$, following the user's post. The first model, $\mathcal{M}^1_c$ (see Fig. 3) captures comments and likes from both contact $c$ and the user following some post. This behaviour is assumed to be independent of a post's sensitivity (as discussed later). States in $\mathcal{M}^1_c$ are: $s_1$, $s_2$, $s_3$ and $s_4$ respectively representing likes by $c$ and user, and comments by $c$ and user; initial state $s_0$; terminal state $s_{\text{end}}$; and $s_6$ (which simply improves readability). Symbolic parameters in $\mathcal{M}^1_c$ are $p_1, p_2, p_3, p_4 \in [0, 1]$ and $r_1, r_2, r_3, r_4 > 0$. Associated transition probabilities, rewards and propositional labels are shown in the figure. For example, if $\mathcal{M}^1_c$ is in state $s_0$, then $p_1$ is the probability that the next reaction is a like by $c$, and given any of the four reactions, the system returns to $s_0$ with certainty.

There are 5 remaining models, $\mathcal{M}^2_{c,l}$ (see Fig. 4), one per sensitivity level $l \in \{0, 1, 2, 3, 4\}$, which capture the reshare behaviour of the contact in response to a post at sensitivity $l$. States in $\mathcal{M}^2_{c,l}$ are: initial state $s_0$; terminal state $s_{\text{end}}$; and $s_5$ representing a reshare by $c$. $\mathcal{M}^2_{c,l}$ has two symbolic parameters $q_l \in [0, 1]$ and $r_5 \geq 0$. Again, transition probabilities, rewards and labels are shown in the figure. In an example execution of $\mathcal{M}^2_{c,l}$ from state $s_0$, $q_l$ is the probability that a $c$ reshares, and this can happen at most once.

We wish to estimate the social benefit of all reactions which follow the user sharing a post at sensitivity $l$ with $c$, and equate this with the total reward accumluated over the lifetime of the parallel execution of $\mathcal{M}^1_c$ and $\mathcal{M}^2_{c,l}$. This corresponds to the sum of the expressions returned by PRISM when $\mathbf{R}_{=?}[\mathbf{F}\text{end}]$ is invoked on each model. These queries can be invoked once at design time and respectively give:

$$V^1_c = \frac{\sum_{i=1}^4 p_i r_i}{(1 - \sum_{i=1}^4 p_i)} \tag{1}$$

$$V^2_{c,l} = q_l r_5 \tag{2}$$

We discuss the use of these expressions in Section V-A.

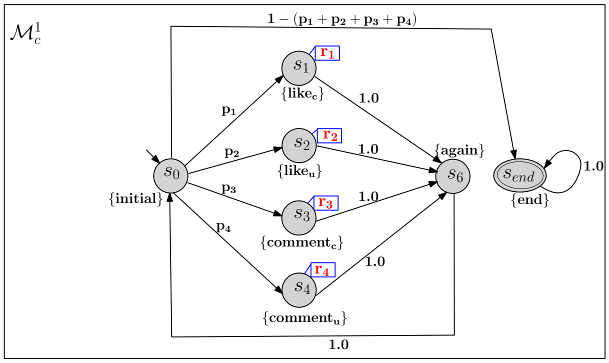

Fig. 3: PMC model of online social interactions between a Facebook user and their post recipient $c$.

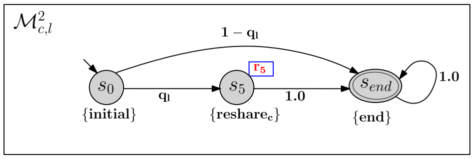

Fig. 4: PMC model capturing the re-sharing behaviour of a recipient $c$ for posts received with sensitivity level $l$.

### C. Online Learning of Sharing Behaviour

PMC modelling and analysis, like that described above, is conventionally used for offline analysis of system properties [15], [16]. We instead apply these techniques at runtime updating our PMC parameters in light of observed interactions. We must, however, consider two complicating factors. First, $c$'s behaviour may change with time, and so we use a variant of the adaptive Bayesian learning algorithm from [17]. Secondly, we can never observe when the user-contact pair stop interacting on a given post. To address this, we account for unobserved transitions to terminal states, $s_{\text{end}}$, by noting that each execution must eventually make this transition, and constructing a synthetic observation to the *end* state for every shared post with the same time-stamp as the original share.

Here, we describe how transition probabilities $p_{ij}$ of a PMC can be learned when the analysed system is operational, and its state transitions monitored (for Facebook these correspond to the monitored events, such as comments, likes, share and reshare). More formally, suppose that, we have observed $K > 0$ transitions out of $s_i \in S$ and that the $k$-th such transition $1 \leq k \leq K$, is to state $s_{j_k} \in S$, we define

$$\sigma^k_{ij} = \begin{cases} 1 & \text{if } j_k = j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

[1] Similar model checkers include MRMC [10] and Ymer [11].

and estimate probability of a state transition from $s_i$ to $s_j$ as

$$p_{ij}^K = \frac{c_i^o}{c_i^o + K} p_{ij}^o + \frac{K}{c_i^o + K} \frac{f_{ij}^K}{g_{ij}^K} \tag{4}$$

where $p_{ij}^0$ is the prior for $p_{ij}^K$, $c_i^0 > 0$ quantifies our confidence, $f_{ij}^1 = \sigma_{ij}^1$, $g_{ij}^1 = 1$, and for $K > 1$, $f_{ij}^K$ and $g_{ij}^K$ are defined recursively as:

$$\begin{aligned} f_{ij}^K &= \alpha^{-(t_K - t_{K-1})} f_{ij}^{K-1} + \sigma_{ij}^K \\ g_{ij}^K &= \alpha^{-(t_K - t_{K-1})} g_{ij}^{K-1} + 1 \end{aligned}$$

Here, $t_K$ represents the timestamp of the $K$-th observation, and $\alpha \geq 1$ is an ageing parameter (see [17] for a full description and proof). These probability estimates have two key features: older observations are downweighted, allowing rapid adaptation to changes in $p_{ij}$; and the recursive form means storage and computation complexity are both in $O(1)$.

### D. Interaction Monitoring

The *Monitor* component detects events that are being tracked by our behavioural models (see Section IV-B), and, when detected, notifies the *Learning Engine* so that appropriate updates can be invoked (see Section IV-C). In order to detect cross-posting based on a post's modality (e.g., text, image, audio or video), in the *Monitor* component, state-of-the-art information retrieval techniques [18] can be implemented as a pluggable module to measure the similarity between posts sent to $c$, and posts $c$ subsequently shares with others. Posts that exceed some similarity threshold (e.g., [19]) can be treated as cross-posts and the *Learning Engine* notified. Such a task is becoming feasible recently with the advance of high performance hardware and also information retrieval techniques [19]–[25].

## V. Decision Support System

In this section, we present the technical details of our decison support system (DSS) that elicits and models the sharing preferences of the user, and informs the user of any sharing decisions that represent significant benefits or exceed certain risk thresholds derived from these preferences. Section V-A and Section V-B formally define the *social benefit* and privacy risk models, and Section V-C describes how we initialise the parameters associated with these models. Finally, in Section V-D we discuss the mechanisms for the user to adjust the parameters of the privacy risk model, based on continual feedback about the working system from the user.

### A. Social Benefit

The DSS estimates the social-benefit a user expects to gain when sharing a post with contact, $c$, from the *Concrete Interaction Models*, described in Section IV-B. For our Facebook example, this is captured by models $\mathcal{M}_c^1$ and $\mathcal{M}_{c,l}^2$. These two models attribute socially-beneficial rewards to relevant social events: reshares (of non-sensitive posts), comments and likes by contact $c$; and comments and likes in response by the user. A user may value each event differently, but for simplicity we assign a fixed value to each event type. As discussed in Section IV, the expected social-benefit for sharing a post with at sensitivity $l$ with contact $c$, is the expected total reward accumulated over the lifetime of the parallel computation of models $\mathcal{M}_c^1$ and $\mathcal{M}_{c,l}^2$, and is given by

$$B_{c,l} = V_c^1 + V_{c,l}^2 = \frac{\sum_{i=1}^4 p_i r_i}{(1 - \sum_{i=1}^4 p_i)} + q_l r_5 \tag{5}$$

For each potential sharing decision, the expected social benefit, $B_{c,l}$, is compared with a threshold, $\bar{B}$, set for the user. If it exceeds that threshold, $B_{c,l} > \bar{B}$, then the user is notified by placing contact $c$ in the *super friends* list. If a different behavioural model were implemented, then the above expression would need to be updated appropriately.

While a variety of choices could be made about rewards in our Facebook model, we suggest the following simple, intuitive choice. Without loss of generality, a contact's comment is given a unitary reward, i.e. $r_3 = 1$. Other rewards $r_1 = \frac{1}{2}, r_2 = \frac{1}{4}, r_4 = \frac{1}{2}$ and $r_5 = 1$ are based on a small study where participants were given a questionnaire[2] about expected levels of interaction following a Facebook post.

### B. Privacy Risk

The DSS also estimates privacy risk – a numerical value of undesirability related to risky decisions. By definition, sharing any post at sensitivity $l = 0$ carries no risk (reshares are desired). Sharing at higher sensitivity $l > 0$ with a contact $c$ who has reshare probability $q_{c,l}$, is considered a risky decision with associated risk

$$R_{c,l} = b \log_2(q_{c,l}) + a_l \tag{6}$$

where $2^{a_l}$ is the damage value associated with a privacy breach (a known reshare) at sensitivity level $l$, and $b > 0$ controls how risk averse the user is (their *risk posture*). When the user considers sharing a post of sensitivity $l$ with $c$, the associated risk, $R_{c,l}$, is compared to the user's risk-threshold, $\bar{R}$. The DSS warns the user if $R_{c,l} > \bar{R}$ by placing contact $c$ into the risky list for that post. We define 5 sensitivity levels $l = 0, 1, 2, 3, 4$. For $l > 0$, $a_l = l$ meaning a privacy breach at sensitivity $l$ is half as damaging as one at $l' = l + 1$ ($l = 0$ means no risk).

Risk posture, $b$, is set so $R_{c,l}$ values, as closely as possible, reflect a user's preferences over risky decisions. Risk appetite, $\bar{R}$, represents the user's maximum acceptable risk. These parameters are given initial values based on user input (Section V-C), then adjusted at runtime by the user (Section V-D).

### C. Initialising Decision Support Parameters

As indicated, initial values for social benefit threshold, $\bar{B}$, risk-posture, $b$, and risk-threshold, $\bar{R}$, are elicited from users via a short, non-technical questionnaire, where they consider outcomes in three sharing scenarios[2]. However, as a user may not feel able to respond to one or more questions, we provide null response options for each scenario, and generate default values based on previous responses.

[2] The questionnaire can be found at bit.ly/2x4Fqqs

The social-benefit threshold, $\bar{B}$ aims to identify contacts who (on average) exceed a user's expected level of social engagement. We quantify this as the expected number of likes, $N_l$, and comments, $N_c$, in response to a typical post from the user to 10 recipients, and use this to predict the *expected social reward per person* as:

$$\bar{B} = 0.1 \cdot (N_c + 0.5 N_l) \tag{7}$$

The risk-posture, $b$, controls how the *probability* of a privacy breach affects the associated risk. Values of $0 < b < 1$ model risk-averseness, $b = 1$ models risk-neutrality and $b > 1$ models risk-seeking behaviour.

We elicit, $b$, indirectly from the user in terms of what we call the trade-probability, $\tilde{q}$ – the probability at which the user would trade exposure to two simultaneous privacy breaches for a guaranteed privacy breach, where all such privacy breaches are considered equally damaging. Risk posture is then calculated as:

$$b = \frac{-1}{\log_2(\tilde{q})} \tag{8}$$

The risk-threshold, $\bar{R} < 0$ (in conjuction with $b$) controls the regularity of warnings in the DSS. This value is again indirectly elicited, this time via the *sensitivity* 1 *trigger probability*, $\bar{q}_1$ – the lowest probability of reshare for which the user would liked to be warned. The risk threshold is then calculated as:

$$\bar{R} = a_1 + b \log_2(\bar{q}_1) = b \log_2(\bar{q}_1) \tag{9}$$

*Trigger probabilities* at other sensitivity levels ($l > 0$) can then be calculated as: $\bar{q}_l = 2^{\frac{\bar{R} - a_l}{b}}$.

### D. Adjusting Decision Support Parameters

To provide additional user control over decision support, and to allow for poorly initialised values to be corrected for, we provide a mechanism to adjust risk-posture, $b$, and risk-threshold, $\bar{R}$, based on repeated contemporaneous judgements of the working system.

*Adjusting $\bar{R}$:* The lowest relevant sensitivity level, $l = 1$, has a maximum associated reshare risk of 0 (see Equation (6)). Therefore $\bar{R}$ must be strictly negative, $\bar{R} < 0$. Otherwise, the user would never be warned at $l = 1$. We therefore propose multiplicative step adjustments to $\bar{R}$ with a fixed factor $\eta > 1$ to lower $\bar{R}$, and $\eta^{-1}$ to raise it. More precisely, if the user indicates that they have too many warnings, then we increase the threshold with: $\bar{R} \leftarrow \eta^{-1} \bar{R}$, which increases $\bar{q}_l$ for all $l \geq 1$. Conversely, if the user indicates they are getting too few warnings then we reduce the threshold with: $\bar{R} \leftarrow \eta \bar{R}$.

*Adjusting $b$:* The risk-posture is also strictly positive, i.e. $b > 0$, so we again propose multiplicative changes by a new step factor, $\zeta > 1$. In our system, an increased risk-posture corresponds to a greater differentiation between sensitivity levels and vice versa. Therefore, the user can increase differentiation between sensitivity levels with an incremental increase in risk-averseness, effected with: $b \leftarrow \zeta b$. Similarly, a user can decrease this differentiation by decreasing risk-averseness, effected with: $b \leftarrow \zeta^{-1} b$. However, to ensure the baseline trigger probability $\bar{q}_1$ remains unchanged, we also adjust the privacy threshold. So an increase in risk-averseness is accompanied by a reduction to the risk-threshold of: $\bar{R} \leftarrow \zeta \bar{R}$. Similarly, decrease in risk-averseness is accompanied by an increase to the risk-threshold of: $\bar{R} \leftarrow \zeta^{-1} \bar{R}$.

With these tools the user can incrementally shape the DSS system to suit their privacy preferences.

## VI. Related Work, Conclusion and Future Work

A considerable body of research has been devoted to address the information sharing problem raised by the increasing number of privacy incidents and regrets happening in OSNs [8], [26]–[34]. However, these approaches do not consider the situation when users may have made poor sharing decisions in the past. Whereas, Machine learning and statistical inference approaches like [35]–[40] study information diffusion in OSNs in order to predict the temporal dynamics of the diffusion process. A very recent work [41] uses Inductive logic programming to build a formal model that learns users' dynamic social identities at runtime in order to analyse group processes and intergroup relations in OSNs.

This paper presents a *learning to share* approach that enables software engineers/developers to readily add adaptive decision support to social network applications, such that the user has fine grained, informed control over their privacy settings. This allows users to maximise their social benefit, whilst controlling risk of privacy breaches to levels they find personally acceptable. We show how our approach can be used in the context of Facebook as the selected OSN platform. However the approach is applicable to other OSNs such as LinkedIn and Google+. This approach could also be readily extended to provide privacy control for a broader family of online interactions, such as undesirable cross-posting behaviours in social question answering services (e.g., StackOverflow) [42].

As for future work, we plan to conduct an online questionnaire in which users are asked to consider outcomes of their online sharing, based on scenarios discussed in Section V-C. This will help us to initialise the parameters of the DSS with realistic values. This will be followed by a user study to help us evaluate the feasibility of our approach from users' perspective within the context of Facebook. For this purpose, a Facebook plug-in has been implemented and we are at the stage of designing and conducting the user study. Finally we will organise a workshop with OSN developers and present our *learning to share* architecture, implemented as Facebook plugin and the user study results. During this workshop we will collect qualitative feedback from developers and elicit their tendency towards integrating our approach as a privacy management for OSNs.

## Acknowledgment

We would like to thank EPSRC, SFI and the ERC for their financial support.

## REFERENCES

[1] A. Lenhart *et al.*, *Social networking websites and teens: An overview*. Pew/Internet, 2007.

[2] C. Dwyer *et al.*, "Trust and privacy concern within social networking sites: A comparison of facebook and myspace," *AMCIS 2007 proceedings*, p. 339, 2007.

[3] K. Subrahmanyam *et al.*, "Online and offline social networks: Use of social networking sites by emerging adults," *Journal of applied developmental psychology*, vol. 29, no. 6, pp. 420–433, 2008.

[4] B. Debatin *et al.*, "Facebook and online privacy: Attitudes, behaviors, and unintended consequences," *Journal of Computer-Mediated Communication*, 2009.

[5] A. Fitzpatrick, "Study says facebook privacy concerns are on the rise - is it accurate?" Mashable, 4 May 2012, 2012.

[6] M. Johnson *et al.*, "Facebook and privacy: it's complicated," in *Proceedings of the eighth symposium on usable privacy and security*. ACM, 2012, p. 9.

[7] M. Madejski *et al.*, "A study of privacy settings errors in an online social network," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*. IEEE, 2012, pp. 340–345.

[8] M. Yang *et al.*, "Adaptive sharing for online social networks: A trade-off between privacy risk and social benefit," in *Proceedings of the 13th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2014, pp. 45–52.

[9] Z. Tufekci, "Can you see me now? audience and disclosure regulation in online social network sites," *Bulletin of Science, Technology & Society*, vol. 28, no. 1, pp. 20–36, 2008.

[10] J.-P. Katoen *et al.*, "A markov reward model checker," in *Quantitative Evaluation of Systems, 2005. Second International Conference on the*. IEEE, 2005, pp. 243–244.

[11] H. L. Younes, "Ymer: A statistical model checker," in *International Conference on Computer Aided Verification*. Springer, 2005, pp. 429–433.

[12] M. Kwiatkowska *et al.*, "Prism 4.0: Verification of probabilistic real-time systems," in *International Conference on Computer Aided Verification*. Springer, 2011, pp. 585–591.

[13] H. Hansson and B. Jonsson, "A logic for reasoning about time and reliability," *Formal aspects of computing*, vol. 6, no. 5, pp. 512–535, 1994.

[14] F. Ciesinski *et al.*, "On probabilistic computation tree logic," in *Validation of Stochastic Systems*. Springer, 2004, pp. 147–188.

[15] C. Ghezzi and A. M. Sharifloo, "Model-based verification of quantitative non-functional properties for software product lines," *Information and Software Technology*, vol. 55, no. 3, pp. 508–524, 2013.

[16] T. Chen, M. Kwiatkowska, D. Parker, and A. Simaitis, "Verifying team formation protocols with probabilistic model checking," in *International Workshop on Computational Logic in Multi-Agent Systems*. Springer, 2011, pp. 190–207.

[17] R. Calinescu *et al.*, "Using observation ageing to improve markovian model learning in qos engineering," in *Proceedings of the 2Nd ACM/SPEC International Conference on Performance Engineering*, ser. ICPE '11. New York, NY, USA: ACM, 2011, pp. 505–510. [Online]. Available: http://doi.acm.org/10.1145/1958746.1958823

[18] M. S. Lew *et al.*, "Content-based multimedia information retrieval: State of the art and challenges," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 2, no. 1, pp. 1–19, Feb. 2006.

[19] R. Baeza-Yates and B. Ribeiro-Neto, *Modern information retrieval*. ACM Press Books, 1999.

[20] A. Huang, "Similarity measures for text document clustering," 2008.

[21] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive datasets*. Cambridge University Pess, 2014.

[22] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[23] V. Patraucean, A. Handa, and R. Cipolla, "Spatio-temporal video autoencoder with differentiable memory," *CoRR*, vol. abs/1511.06309, 2015. [Online]. Available: http://arxiv.org/abs/1511.06309

[24] G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors," in *Proceedings of the Fourth ACM International Conference on Multimedia*, 1996, pp. 65–73.

[25] A. McCallum, K. Nigam, and L. H. Ungar, "Efficient clustering of high-dimensional data sets with application to reference matching," in *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000.

[26] S. Wilson *et al.*, "Privacy manipulation and acclimation in a location sharing application," in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2013, pp. 549–558.

[27] Y. Wang *et al.*, ""I Regretted the Minute I Pressed Share": A Qualitative Study of Regrets on Facebook," in *Proceedings of the Seventh Symposium on Usable Privacy and Security*, ser. SOUPS '11. ACM, 2011, pp. 10:1–10:16.

[28] T. Dinev *et al.*, "An extended privacy calculus model for e-commerce transactions," *Information System Research*, vol. 17, no. 1, pp. 61–80, 2006.

[29] H. Krasnova *et al.*, "Online social networks: Why we disclose," *Journal of Information Technology*, vol. 25, no. 2, pp. 109–125, 2010.

[30] H. Xu *et al.*, "The role of push-pull technology in privacy calculus: The case of location-based services," *Journal of Management Information Systems*, vol. 26, no. 3, pp. 135–174, 2009.

[31] H. Nissenbaum, "Privacy as Contextual Integrity," *Washington Law Review*, vol. 79, no. 1, 2004.

[32] A. Barth *et al.*, "Privacy and contextual integrity: Framework and applications," in *IEEE Symposium on Security and Privacy*, 2006, pp. 184–198.

[33] Y. Krupa *et al.*, "Handling privacy as contextual integrity in decentralized virtual communities: The privacias framework," *Web Intelli. and Agent Sys.*, vol. 10, no. 1, pp. 105–116, 2012.

[34] I. Bilogrevic *et al.*, "Adaptive information-sharing for privacy-aware mobile social networks," in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2013, pp. 657–666.

[35] J. Yang *et al.*, "Modeling information diffusion in implicit networks," in *Proceedings of the 2010 IEEE International Conference on Data Mining*, 2010, pp. 599–608.

[36] J. Leskovec *et al.*, "Meme-tracking and the dynamics of the news cycle," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 497–506.

[37] A. Guille *et al.*, "A predictive model for the temporal dynamics of information diffusion in online social networks," in *Proceedings of the 21st International Conference on World Wide Web*, 2012, pp. 1145–1152.

[38] S. Huang *et al.*, "Predicting aggregate social activities using continuous-time stochastic process," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, 2012, pp. 982–991.

[39] T. A. B. Snijders, "The statistical evaluation of social network dynamics," 2001.

[40] J. Li *et al.*, "Social network user influence dynamics prediction," in *Web Technologies and Applications*, 2013, pp. 310–322.

[41] G. Calikli *et al.*, "Privacy Dynamics: Learning Privacy Norms for Social Software," in *International Symposium on Adaptive and Self-Managing Systems*, 2016.

[42] B. S. Butler and X. Wang, "The cross-purposes of cross-posting: Boundary reshaping behavior in online discussion communities," *Information Systems Research*, vol. 23, no. 3-part-2, pp. 993–1010, 2012.