2019 IEEE International Conference on Services Computing (SCC)

Helping Users Managing Context-based Privacy Preferences

Md. Zulfikar Alom
DiSTA, University of Insubria
Varese, Italy
mzalom@uninsubria.it

Barbara Carminati
DiSTA, University of Insubria
Varese, Italy
barbara.carminati@uninsubria.it

Elena Ferrari
DiSTA, University of Insubria
Varese, Italy
elena.ferrari@uninsubria.it

***Abstract*—Today, users interact with a variety of online services offered by different providers. In order to supply their services, providers collect, store and process users' data according to their privacy policies. To have more control on personal data, user can specify a set of privacy preferences, encoding the conditions according to which his/her data can be used and managed by the provider. Moreover, many services are context dependent, that is, the type of delivered service is based on user contextual information (e.g., time, location, and so on). This makes more complicated the definition of privacy preferences, as, typically, users might have different attitude with respect the privacy management based on the current context (e.g., working hour, free time). To provide a more fine-grained control, a user can set up different privacy preferences for each different possible contexts. However, since user change the context very frequently, this might result in a very complex and time-consuming task. To cope with this issue, in this paper, we propose a context-based privacy management service that helps users to manage their privacy preferences setting under different contexts. At this aim, we exploit machine learning algorithms to build a classifier, able to infer new privacy preferences for the new context. The preliminary experimental results we have conducted are promising, and show the effectiveness of the proposed approach.**

***Keywords*—Context awareness; privacy preferences; learning; context-based services.**

I. Introduction

In recent years, users interact with many different services managed by a variety of providers. In order to supply their services (e.g., entertainment, health monitoring, etc.), providers collect, store and process a massive amount of personal information about users. Every service provider has its own privacy policies, which express how it collects and manages individuals' personal information. In contrast, to improve the privacy control, individuals can explicitly express their privacy preferences, which state the conditions according to which their data have to be used and managed. Today, most of the provided services are *context dependent*, that is, the deliver of the service is based on user contextual information (e.g., time, location, and so on). Examples are location-based services, IoT based services, and so on. Contextual information refers to any piece of data of the individual that can be used to define his/her current situation. Typically, contexts can impact the user privacy preferences, for instance, a user may feel comfortable to access *entertainment* services when (s)he stays at home, but (s)he will not be comfortable to access the same type of services during office hours, when he/she is in his/her office. Many studies show how contextual information is important in privacy preferences specification. For instance, Nissenbaum et al. [1] show that most of the privacy preference models fail to protect against violations of user privacy preferences because they do not keep into account contextual information. As a matter of fact, many of the existing privacy preferences frameworks (e.g., [2], [3], [4]) do not consider individuals' contextual information to make privacy aware decisions.

To cope with this limitation, users might specify context-based privacy preferences, that is, privacy preferences stating conditions on how personal data has to be used based on current situation (e.g., no access to *entertainment* services when the location is office). This brings the nice benefit of increasing the user control over his/her data. However, since a user may interact with several contexts, it also increases the number of preferences that (s)he has to specify and manage, resulting in a very complex and time-consuming task. For this reason, in this paper, we propose a service that helps users to manage their privacy preferences when they move to a new context. More particularly, we design a framework that infers individuals' privacy preferences based on their contextual information.

The overall idea is that when a user enters in a new context CTX_{new}, the service automatically sets a new privacy preference for it, by leveraging on user previously specified existing context-based privacy preferences. More precisely, among the privacy preferences, the proposed approach identifies the preferences defined for contexts that are "similar" to CTX_{new}. Then, it exploits them as a baseline to define the new privacy preference for CTX_{new}. The above-described process has been designed such as to take into account users' privacy perspective. That to say, we do not want to generate a new privacy preference that, even if related to those defined for similar contexts, relaxing some conditions that are relevant for a given user. At this purpose, we propose to learn from a user two crucial aspects. The first is about which fields of a context (e.g., time, location, etc.) are considered more informative in setting his/her privacy preferences by the user. This information is useful to identify among the contexts for which the user has already defined his/her preferences, the one

2474-2473/19/$31.00 ©2019 IEEE
DOI 10.1109/SCC.2019.00027

that is most relevant to CTX_{new}. The second information to be learnt is how much the user selected change the privacy preference associated with the most relevant context, in order to adapt it for CTX_{new}.

At this purpose, we exploit machine learning algorithms, to build a classifier able to decide which privacy preference components a user is pron to modify (e.g., purpose, retention) and the corresponding update range, as well as, which are the context components that trigger the preference update. More precisely, we take into account user feedback to create a training dataset on which learning algorithms build the classifiers. The learned classifiers are then used to derive new privacy preferences for the new context.

To show the feasibility of the proposed approach, we have conducted some preliminary experiments. In particular, we compared the proposed approach with a naive approach, that is, a classifier suggesting context-based privacy preferences without leveraging on previously specified context-based privacy preferences. At this purpose, we have extended the approach in [4], where a learning approach has been proposed to suggest *non context-based* user privacy preferences. The approach in [4] first creates a training dataset of user labels on a set of service requests (e.g., a label is the accept/deny decision on a given service request), then generate a classifier on it, able to automatically decide if a new service request has to be accepted or denied.[1] In order to compare the proposed approach with [4], we extended the latter such as to collect users' labels on service requests complemented with context information. Moreover, we test different supervised machine learning algorithms, namely, Logistic Regression, Random Forest, and Naive Bayesian [5]. The obtained results show that the proposed approach gives better performance than the naive approach.

The rest of this paper is organized as follows. Section II discusses related work. In Section III, we describe our proposal, whereas Section IV shows the results of our preliminary experiments. Finally, Section V concludes the paper.

II. Related work

In the literature, many papers have addressed context awareness to dynamically adapt users' privacy preferences. For instance, Behrooz et al. [6] proposed a context-aware privacy policy language (CPPL) that uses context to pre-filter policies applicable to the current situation, in order to reduce the number of policies that actually have to be evaluated. Konings et al. [7] proposed a context-aware privacy policy selection model, exploiting user's current location and other potential contextual features, such as time, user's activities, or mood. Bunnig et al. [8] proposed an abstract disclosure decision model and argued that an appropriate context abstraction is required to match the users privacy preferences.

Some approaches implemented context-based privacy preferences in the smart-phone environment. For example, Wijesekera et al. [9] proposed a novel privacy management system that relies on user's contextual information, to improve user privacy decision making capability in mobile platforms; whereas [10] proposed user's location sharing privacy preferences by considering contextual information. Yuan et al. [11] proposed a privacy-aware model for photo sharing based on machine learning by exploiting contextual information. The proposed model utilizes image semantics and requester contextual information to decide whether or not to share a particular picture with a specific requester in a certain context. Likewise, [12] proposed a privacy preference framework that semi-automatically predicts sharing decision, based on personal and contextual features.

However, the approach we propose in this paper differs from all the above-mentioned proposals in that none of the above works support user in the complex task of specification of context-based privacy preferences.

III. Proposed method

As introduced in section I, individuals' privacy expectations are highly context dependent, and since users change their context very often, setting up privacy preferences for every new context is a very complex and time-consuming task. To address this issue, we design a service helping users to manage privacy preferences setting when they move from one context to another. The main idea is to infer the best privacy preferences for the new context leveraging on privacy preferences previously specified by the user for different contexts. Thus, as a first step, we need to select among the contexts for which the user has already specified a preference, those that are similar to the new one. In doing this, we do not simply rely on a similarity measure between contexts, but we also want to keep into account users' perspective (see Section III-C). Indeed, given a similarity measure, two existing contexts could have the same distance to the new one but differ on a few fields (e.g., time, location) that are very relevant for that user. As such, we would like also to take into account, for a target user, his/her preferences on which context field is more informative and thus should have more relevance in the similarity measure.

Once the most similar and most relevant context has been selected, the system has to retrieve the corresponding privacy preference. Here, the basic assumption is that this preference might represent a good match for the new context but some slightly modifications might be needed as well. In order to understand whether and how the identified privacy preference has to be adapted for the new context, we want to take into account once again user's perspective. More precisely, to learn which fields of the identified privacy preference need

[1] Due to space limitations, we do not provide in this paper details about how the learning approach works, by referring the interested reader to [4] for more details.

to be modified (e.g., purpose, data, retention, and recipient), we exploit machine learning to infer, from each user, which component and how is willing to adapt it.

All the above described steps will be described in the following, starting from the modeling of contexts and context-based privacy preferences.

A. Contexts and context-based privacy preferences modeling

A context is defined by the information used to characterize the present situation of individuals (e.g., activity, time, etc.). For the sake of simplicity, in this paper, we consider contexts containing information on four different dimensions, namely, time, location, activity, and social. The latter two represent the action the user is currently doing (e.g., work, run, relaxing, etc) and companion with whom he is (e.g., alone, friends, colleagues, etc). However, the approach can be extended to the consideration of additional contextual information.

More formally, user contexts can be defined as follows.

*Definition 1: (***User context***). A context for a user U, denoted as CTX_U, is a set of pairs $\{(tm, v_{tm}), (lc, v_{lc}), (ac, v_{ac}), (sl, v_{sl})\}$, where the first component is a contextual property and the second its corresponding value. More precisely, tm denotes a time, lc a user location, ac denotes a user activity, whereas sl specifies the social dimension (i.e., user companion).*

Example 1: Let us suppose that a user *U*'s $location$ is $library$, time is $Monday\ morning$, and (s)he is $studying$ with his/her $brother$. Therefore, the current context of *U* can be modelled in the following way: $CTX_U = (Monday\ morning, library, studying, brother)$

To provide a more fine-grained control of personal data release, a user can set up different privacy preferences for different contexts, stating the conditions according to which his/her data has to be used and managed in that particular context. We formally define a user's context-based privacy preference as follows.

*Definition 2: (***Context-based privacy preference***). A context-based privacy preference for a user U, denoted as CTX_{pp_U}, is a tuple (CTX, PP), where, CTX is a context, and PP is a tuple (p, d, ret, rec), where, p denotes the purpose for which a service provider is allowed to collect the data denoted by d, ret specifies how long the service provider can store the data, whereas rec indicates whether additional third party entities can use the data.*

Example 2: Let us consider a user *U* that wishes to release his/her $name$, $date\_of\_birth$, $certificates$ data only for $admission$ purpose. Moreover, (s)he wants that the data will not be retained more than $260\ days$, allowing service providers to share it with $third\_party$. Let us assume that the user wants this preference to be enforced in the context presented in Example 1. Therefore, such context-based privacy requirements can be encoded through the following context-based privacy preference: $CTX_{pp\_U} = (\{Monday\ morning, library, studying, brother\}, \{admission, name, date\_of\_birth, certificates, 260\ days, yes\})$

B. Context distance metrics

As mentioned earlier, when a user moves to a new context, we calculate a similarity score between the new context and all the contexts for which the user has already defined a privacy preference. To find the most similar context and corresponding privacy preference, we measure a distance to determine how far the new context is from existing contexts. To do so, we measure the distance of each context component, as explained in what follows.

Time distance: time can be expressed as a numerical value[2], hence, we can use the *Euclidean distance* [13] to measure the time distance between different contexts.

*Definition 3: (***Time distance***). Let CTX_{U_n} be a user new context, and CTX_{U_p} be a user's prior context. Let $max(CTX_{U_n}.tm, CTX_{U_p}.tm)$ be the maximum value between the time components. Therefore, the time distance is defined as follows:*

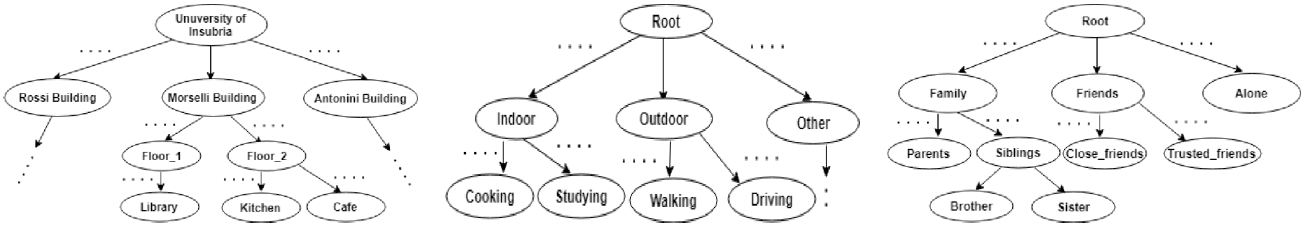
$$D_{tm}(CTX_{U_n}.tm, CTX_{U_p}.tm) = \frac{|CTX_{U_n}.tm - CTX_{U_p}.tm|}{max(CTX_{U_n}.tm, CTX_{U_p}.tm)}$$

Location distance: In this work, rather than the exact GPS location, we are interested in modelling locations that can be sensitive for personal data release (e.g., home, office). To this end, we rely on the Aura Location Identifier (ALI) model [14]. The main idea of this model is to decompose physical spaces into different levels of spaces. For instance, the campus of the *University of Insubria* can be decomposed into several spaces: *Rossi Building, Morselli Building, Antonini Building, etc*. Each of these buildings is in turn divided into smaller composing sub-spaces, until reaching enough precision. This hierarchical representation is called a *space tree*, where each node corresponds to a given space in the physical environment. Figure 1a shows part of a space tree (i.e., location hierarchy) for the *University of Insubria*. By exploiting this hierarchy, we measure the distance between spaces, by leveraging on the *Wu and Palmer similarity* [15] metric.[3]

*Definition 4: (***Location distance***). Given two location l_1 and l_2, let ccn be the closest common ancestor between l_1 and l_2 in the space tree, $depth(ccn)$ be the number of edges from the root to ccn, $dis(l_1)$ and $dis(l_2)$ be the distance*

[2]For the sake of simplicity, in this paper, the time is expressed by only considering 4 time slots for each week day (e.g., Monday morning, Monday night, etc.).

[3]Note that alternative similarity metrics can be easily used as well.



(a) Location hierarchy (b) Activities hierarchy (c) Social hierarchy

Figure 1: Location, activity, and social hierarchies

of location l_1 and l_2 from ccn, respectively. Therefore, the location distance is defined as follows:

$$D_{lc}(l_1, l_2) = 1 - \frac{2 * depth(ccn)}{dis(l_1) + dis(l_2) + 2 * depth(ccn)}$$

Activity distance: we exploit the Hierarchical Structural Model (HSL-model) [16] to represent individuals' daily activities, being able to build an *activity ontology* (see Figure 1b). If a_1 and a_2 are two activities, we measure their distance, denoted as $D_{ac}(a_1, a_2)$, as in Definition 4, by exploiting the activity hierarchy instead of the location one.

Social distance: to calculate the distance between the social attributes, we exploit an ontology similar to the one used by social networks [17]. If s_1 and s_2 are two social attributes, we measure their distance, denoted as $D_{sl}(s_1, s_2)$, as Definition 4, by exploiting the social hierarchy (cfr. Figure 1c).

Example 3: Let us consider user U's context-based privacy preference (CTX_{pp_U}) presented in Example 2. Let us assume that U moves to a new context, according to which location is $kitchen$, time is $Monday\ morning$, activity is $cooking$ with $parents$. According to the location, activity, social, time distance definitions and the hierarchies shown in Figure 1, we can measure the distance between each context components as follows:

$$\begin{aligned} D_{lc}(library, kitchen) &= 1 - \frac{(2*1)}{(2+2+2*1)} = 1 - \frac{2}{6} \\ &= 1 - 0.34 \\ &= 0.64 \end{aligned}$$

Likewise,

$$D_{ac}(studying, cooking) = 1 - 0.5 = 0.5$$
$$D_{sl}(brother, parents) = 1 - 0.4 = 0.6$$
$$D_{tm}(Monday\ morning, Monday\ morning) = 0$$

C. Context similarity

When computing the similarity between two contexts, we take into account which are the contextual information that a user considers more relevant in determining the similarity. Therefore, we add weights to the similarity measure to keep into account the user perspective. Such weights will be learnt by using machine learning algorithms (as explained in Section III-D).

*Definition 5: (***Context similarity score***). Let CTX_{U1} and CTX_{U2} be two contexts for user U. Let $w_1, \dots w_4$ be the weights associated with each of the four context attributes. Therefore, the similarity score is defined as follows:*

$$Sim_w(CTX_{U1}, CTX_{U2}) = 1 - \frac{\begin{pmatrix} w_1 * D_{tm}(CTX_{U1}.tm, CTX_{U2}.tm) + w_2 * D_{lc}(CTX_{U1}.lc, CTX_{U2}.lc) \\ + w_3 * D_{ac}(CTX_{U1}.ac, CTX_{U2}.ac) + w_4 * D_{sl}(CTX_{U1}.sl, CTX_{U2}.sl) \end{pmatrix}}{4}$$

Example 4: Let us consider user U's context-based privacy preference presented in Example 2, and the new contexts and related distance measures illustrated in Example 3. Suppose that the learned weights are $w_1 = 0.1$, $w_2 = 0.2$, $w_3 = 0.3$, and $w_4 = 0.4$. Therefore, according to Definition 5, the similarity score is calculated as follows:

$$\begin{aligned} Sim_w(CTX_1, CTX_2) &= \\ 1 - \frac{0.1 * 0 + 0.2 * 0.66 + 0.3 * 0.5 + 0.4 * 0.6}{4} & \\ = 1 - \frac{0 + 0.132 + 0.15 + 0.24}{4} = 1 - 0.13 = 0.87 & \end{aligned}$$

D. Learning mechanism

In this subsection, we explain the learning strategy we have designed. Since we need users feedback, we exploit supervised machine learning algorithms, namely, Logistic Regression (LR), Random Forest (RF), and Naive Bayesian (NB) [5]. Moreover, the preliminary experiments we have carried out show that, with a reasonable user burden (i.e., asking only 30 questions), we are able to get a dataset that is adequate for supervised learning, hence, we do not consider semi-supervised learning approaches.

We use machine learning algorithms to build a classifier able to decide which privacy preference will be set for the user new context. More particularly, we build a classifier able to decide: (i) which elements of the context are more relevant

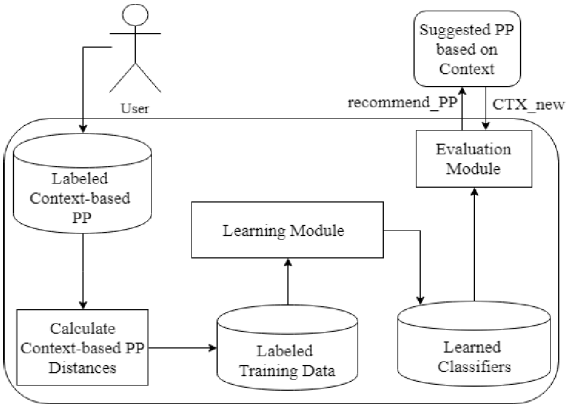


Figure 2: Learning architecture

(e.g., location. time) for the target user; and (ii) how much privacy preference components associated with preferences defined for similar contexts could be modified.

For this purpose, the proposed solution has a first training phase (see Figure 2), where the user is required to judge whether a given privacy preference works for a certain context or (s)he wishes to modify it. More precisely, let CTX_{new} be a new context and CTX_{sim} the most similar context to CTX_{new}.[4] We take the privacy preference associated with CTX_{sim}, denoted as PP_{sim_ctx}, and we ask the user if (s)he would adopt PP_{sim_ctx} in CTX_{new} as is or would like to modify it. In the latter case, we let the user modify it obtaining a new preference, denoted as PP_{mod}. Then, we measure the distance between each component of PP_{sim_ctx} and PP_{mod}, denoted as Dis_{pp}. For measuring the distance between privacy preferences components, we rely on the metrics introduced in our previous work [18]. Finally, the features set on which we build the proposed classifier consists of Dis_{ctx}, Dis_{pp} and the assigned label (adopt/modify), where Dis_{ctx} is the set of distance between each component of CTX_{new} and CTX_{sim}.

Once the learning phase is concluded, we exploit the learned classifiers to infer privacy preferences for new contexts. More particularly, when the user moves to a new context CTX_{new}, the proposed approach (i.e., evaluation module) computes similarity scores between the new context and all contexts for which a privacy preference has been previously specified. We recall that according to Definition 5, this score exploits $w_1, \ldots, w_4$ weights to take into account which context field (e.g., time or location) is more relevant to the user. These weights are initialized with values of weights of the classifier model built in the training phase. As an example, w_1 weight, associated to the time context field in Definition 5, is initialized with the value of weight computed by classifier for the feature containing the distance between time in Dis_{ctx}.

[4]We assume that user has inserted a preliminary small set of context-based privacy preferences. CTX_{sim} is selected among contexts associated to these preferences.

Then, it selects the privacy preference of the most similar context, and adapts the selected privacy preference according to the learnt user's adapting attitudes. More precisely, to determine how much each privacy preference component has to be modified, we exploit the correlation between privacy preference components and the context fields. For this purpose, we used a linear regression model [19], defined as $Y = r_1x_1+r_2x_2+\ldots+r_nx_n+\epsilon$ where, $r_1, \ldots, r_n$ are the regression coefficients, $x_1, \ldots, x_n$ are the independent variables, ϵ is the constant, and n is the number of attributes (our case $n = 4$). To train this model, for each user, we have used the feature set Dis_{ctx} as independent variables and as a target variable/label (i.e., Y) the value of Dis_{pp}. Once trained, this model can be used to update each privacy preference component. For example, let consider, as the privacy preference component that needs to be modified, the retention attribute ret_{mod} having value 260 days. Let assume that the learned coefficients $r_1, \ldots, r_4$ are 2, 8, 15, and 11, respectively, and the constant value ϵ is -10. Let suppose the distance values between the new context and the similar context is the one presented in Example 3. Therefore, the update retention value would be: $ret_{updt} = ret_{mod} + Y = 260 + (2 * 0 + 8 * 0.64 + 15 * 0.5 + 11 * 0.6 - 10) = 270$ days.

IV. Experiments

In this section, we illustrate a series of preliminary experiments we have performed to show the effectiveness of the proposed approach. More particularly, to demonstrate the feasibility of the proposed approach, we compare it with the naive strategy of inferring new context-based privacy preferences from scratch. At this aim, we have extended the approach proposed in [4], where a learning approach has been proposed to suggest traditional (i.e., non context-based) privacy preferences. More precisely, [4] creates a training dataset of user labels on a set of service requests (e.g., accept/deny decision on a given service request), and builds a classifier on this training dataset, able to automatically decide if a new service request should be accepted or denied. To make a fair comparison with the proposed approach, we modify the learning strategy proposed in [4] so that it can infer privacy preferences from service requests containing also contextual data. We first measure the accuracy and F1 score obtained by using LR, RF, and NB. Next, we compute the *satisfaction level* of users regarding privacy preference suggestions generated by both approaches using various learning strategies. In addition, we also evaluate the user quality in terms of feedback on the training dataset to examine how a badly labeled training dataset impacts user satisfaction.

A. Experimental settings

To generate a meaningful dataset, we consider the following contextual information: 23 different locations (e.g., *home, office, university*, and so on), 7 days (e.g., *Sunday, Monday*, etc.) with 4 time slots (e.g., *morning, afternoon,*

		Predicted class	
		Yes	No
True class	Yes	TP_{yes}	$E_{yes,no}$
	No	$E_{no,yes}$	TP_{no}

Table I: Confusion matrix

evening, and night), 12 different user activities (e.g., *studying, meeting, sleeping*, and so on), and 10 different social attribute values (e.g., *alone, family, friends*, and so on). For privacy preferences, we consider: 12 purposes (e.g., *payment, treatment, research*, and so on) for which a service provider is allowed to collect users' data, 30 different data types (e.g., *name, email_id, phone_number, date_of_birth, credit_card, blood_pressure*, and so on) to which privacy preferences can refer to, retention time between 1 to 365 days, and third-party data access status either yes or no.

To collect labels for the training dataset, we developed a web application through which users can give feedback on system-generated privacy preferences. We have recruited two types of evaluators, namely, university-based and crowdsourcing based evaluators. More particularly, we first collected data of 10 CS students from the Islamic University, Bangladesh. Then, to have a bigger group of evaluators with different nationalities and ages, we used the Microworkers crowd-sourcing platform.[5] From this platform, we have collected data from 50 evaluators (aka workers). We recall that to learn user's aptitude in adapting his/her context-based privacy preferences, we posed some questions to the users. The answers to these questions are then used as a labeled training dataset on which we build the classifiers. It should be pointed out that we have collected two types of dataset. First, we collect a labeled training dataset for the naive approach from 30 users (i.e., 10 CS students plus 20 workers), denoted in what follows as naive approach dataset, then we exploit other 30 users to collect a labeled training dataset for evaluating our approach (we name this as proposed approach dataset).

To evaluate whether the classifier correctly works on the new context, we also ask users to give their feedback on the privacy preference suggestions generated by the system (i.e., testing phase).

After collecting the datasets, we have trained the learning algorithms by exploiting the R platform [20]. In order to measure the effectiveness of the proposed approach, we consider the confusion matrix illustrated in Table I. According to this, we exploit the standard evaluation metrics, namely, accuracy, precision, recall, and F1-score, illustrated in Table II.

B. Effectiveness

In this experiment, we carried on a comparative analysis of accuracy and F1-score obtained by the proposed approach and the naive one using different classifiers.

[5]https://www.microworkers.com

Accuracy = $(TP_{yes} + TP_{no})$ / total number of samples
Precision Yes = TP_{yes} / $(TP_{yes} + E_{no,yes})$
Precision No = TP_{no} / $(TP_{no} + E_{yes,no})$
Recall Yes = TP_{yes} / $(TP_{yes} + E_{yes,no})$
Recall No = TP_{no} / $(TP_{no} + E_{no,yes})$
$F1_C$ = $(2 * Precision_C * Recall_C)/(Precision_C + Recall_C)$, where $C \in \{Yes,\ No\}$

Table II: Metrics definition

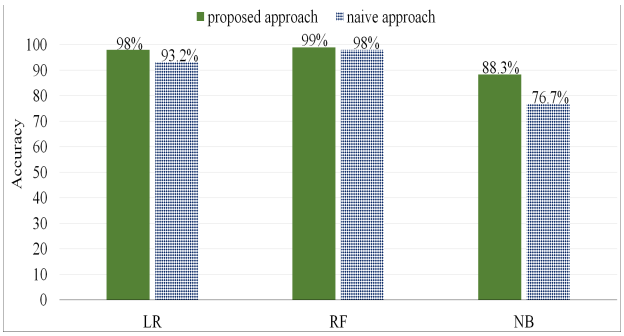


Figure 3: Comparison of accuracy of different approaches

Accuracy. As a first experiment, we use the training datasets and re-label them with the built classifiers. As shown in Figure 3, about 99% and 98% of the proposed approach and naive approach datasets have been correctly labeled by RF, respectively. Likewise, around 98% and 93.2% of the proposed approach and naive approach datasets have been correctly labeled by LR, respectively, whereas, only 88.3% and 76.7% of the two training datasets have been correctly labeled by NB, respectively. Therefore, we can see that RF gives better performance on proposed approach than the naive one.

F1-score. We have calculated the F1 score for each class (yes, no) for comparing the performance among the learning approaches over the training dataset and testing dataset (see Table III). Figures 4 and 5 represent the F1 score comparison for the two approaches using different classifiers. it can be observed that, for both approaches, RF gives greater F1-score. More particularly, by using RF, the proposed approach achieves 97% and 94% F1-score for the class label *yes* on the training and testing dataset, respectively.

C. Participant evaluation

In this experiment, in order to evaluate user satisfaction, we collect feedback from the users regarding the privacy preference suggestions taken by both approaches using various learning strategies.

Satisfaction level. We exploit the developed web application to show evaluators the system-generated privacy preferences for the new context, and we ask the evaluators to give their feedback regarding the system-generated suggestions. More precisely, we have shown to each evaluator 15 context-based privacy preferences, where, 12 of them have been

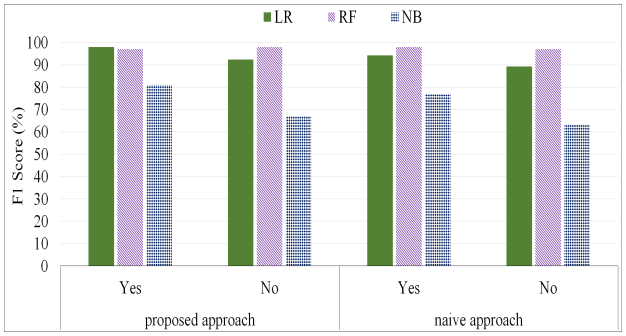


Figure 4: Comparison of F1 score of different classifiers for training dataset

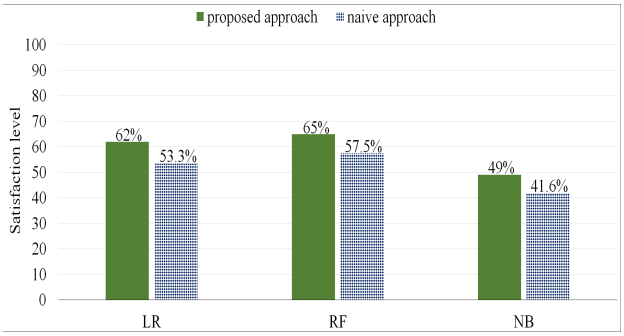


Figure 6: Comparison of evaluators satisfaction level for different approaches

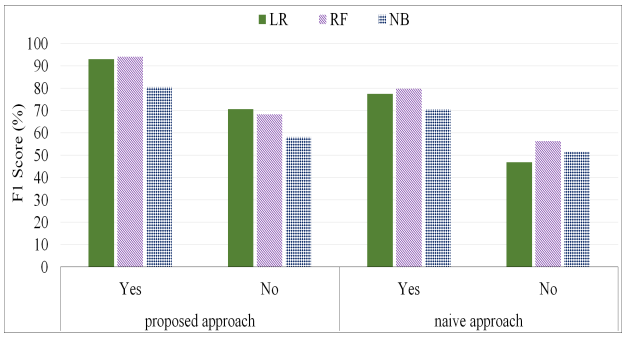


Figure 5: Comparison of F1 score of different classifiers for testing dataset

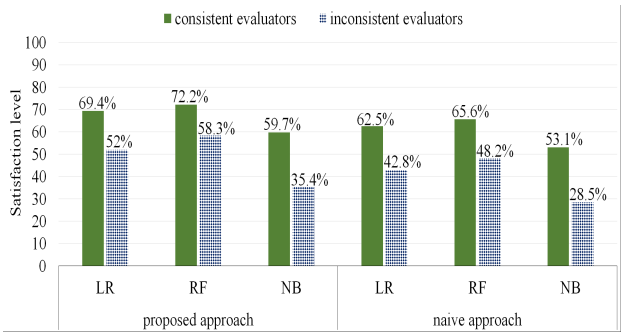


Figure 7: Satisfaction level of consistent and inconsistent evaluators

generated by the classifiers, whereas the remaining 3 are taken from the set of context-based privacy preferences that the evaluators have set up during the learning phase. These are used for checking the consistency of evaluators feedback and measuring the evaluators' quality (as later explained). As shown in Figure 6, about 65% of the evaluators are satisfied with the suggestions given by the proposed approach using RF, whereas, around 57.5% of the evaluators are satisfied with the suggestions given by the naive approach. Similarly, by using LR, around 62% and 53.3% of the evaluators are satisfied with the suggestions given by the proposed approach and the naive approach, respectively. Likewise, by using NB, about 49% and 41.6% of the evaluators are satisfied with the suggestions given by the proposed approach and the naive one, respectively. Therefore, it is clear that the proposed approach achieves higher satisfaction level than the naive one.

Evaluators quality. Through this experiment, we are interested in investigating how a badly labeled training dataset impacts the satisfaction level. With this aim, we used some techniques to identify consistent and inconsistent evaluators. To do so, we have taken 3 context-based privacy preferences from the set of context-based privacy preferences that evaluators have labeled during the training phase. Then, in the testing phase, the web application shows these privacy preferences again to them, and collect the label (i.e., satisfaction level) they assign. Based on this, we can judge whether the evaluator is consistent or not in his/her decisions. We consider that an evaluator is consistent if any two out of three decisions taken during the training and the testing phase match. Figure 7 presents the comparative analysis of the satisfaction level of consistent and inconsistent evaluators for both the approaches. It can be seen that, the satisfaction level of consistent evaluators is greater than the satisfaction level of inconsistent evaluators. However, even in the worst case, about 35.4% of inconsistent evaluators are satisfied with the decisions by the proposed approach, whereas, it is only 28.5% for the naive approach.

However, from the above experimental results, it is clear that the proposed approach provides better performance in terms of accuracy and satisfaction level. Besides, these results let us think that this is a good direction to study and there is room for improvement. By doing more experiments, we need to understand why users are not satisfied with the system-generated suggestions and how we can increase their satisfaction level.

V. Conclusion

In this paper, we have proposed a service for helping users to manage their context-based privacy preferences. To show the feasibility of the proposed approach, we compared the proposed approach with a naive approach, by using three different machine learning algorithms (i.e., LR, RF, and NB).

			LR		RF		NB	
			Yes	No	Yes	No	Yes	No
Training dataset	Proposed approach	Precision	98.1%	96.6%	96.1%	99.5%	83.9%	75.5%
		Recall	99%	90.7%	96.4%	97%	83.8%	64.1%
		F1-score	98%	92.4%	97%	97.8%	81%	66.9%
	Naive approach	Precision	93.4%	93.2%	99%	98%	79.2%	60.6%
		Recall	95.1%	87.2%	98%	97%	75.1%	68.3%
		F1-score	94.2%	89.3%	98%	97%	76.9%	63.2%
Testing dataset	Proposed approach	Precision	91.7%	73.3%	92.3%	68.9%	79.93%	58.8%
		Recall	96.6%	68.9%	95.3%	68%	83.33%	61.4%
		F1-score	93%	70.6%	94%	68.2%	80.56%	58.13%
	Naive approach	Precision	76.4%	52.8%	79.2%	57.6%	73.2%	55.1%
		Recall	80.7%	46.4%	81.8%	57.8%	71%	54.3%
		F1-score	77.5%	46.9%	79.8%	56.3%	70.6%	51.6%

Table III: Performance comparison of different learning algorithms for the training and testing datasets

The experimental results show that the proposed approach provides better performance.

In the future, we plan to conduct more user studies, and to deploy the proposed service into real settings. Also, we plan to investigate other ontology-based distance measures [21] to check how these impact the context selection.

ACKNOWLEDGMENTS

This work has received funding, in parts, from "RAIS: Real-time Analytics for the Internet of Sports", supported by the European Union's Horizon 2020 research and innovation programme under Marie Skodowska-Curie grant agreement No 813162, and from CONCORDIA, the Cybersecurity Competence Network supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 830927.

REFERENCES

[1] H. Nissenbaum, "A contextual approach to privacy online," *Daedalus*, vol. 140, no. 4, pp. 32–48, 2011.

[2] B. C. Singh, B. Carminati, and E. Ferrari, "Learning privacy habits of pds owners," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2017, pp. 151–161.

[3] Y.-A. de Montjoye, E. Shmueli, S. S. Wang, and A. S. Pentland, "openpds: Protecting the privacy of metadata through safeanswers," *PloS one*, vol. 9, no. 7, p. e98790, 2014.

[4] B. C. Singh, B. Carminati, and E. Ferrari, "Privacy-aware personal data storage (p-pds): Learning how to protect user privacy from external applications," *IEEE Transactions on Dependable and Secure Computing*, 2019.

[5] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," 2007.

[6] A. Behrooz and A. Devlic, "A context-aware privacy policy language for controlling access to context information of mobile users," in *International Conference on Security and Privacy in Mobile Information and Communication Systems*. Springer, 2011, pp. 25–39.

[7] B. Könings, F. Schaub, and M. Weber, "Privacy and trust in ambient intelligent environments," in *Next Generation Intelligent Environments*. Springer, 2016, pp. 133–164.

[8] C. Bünnig, "Smart privacy management in ubiquitous computing environments," in *Symposium on Human Interface*. Springer, 2009, pp. 131–139.

[9] P. Wijesekera, J. Reardon, I. Reyes, L. Tsai, J.-W. Chen, N. Good, D. Wagner, K. Beznosov, and S. Egelman, "Contextualizing privacy decisions for better prediction (and protection)," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 2018, p. 268.

[10] J. Xie, B. P. Knijnenburg, and H. Jin, "Location sharing privacy preference: analysis and personalized recommendation," in *Proceedings of the 19th international conference on Intelligent User Interfaces*. ACM, 2014, pp. 189–198.

[11] L. Yuan, J. Theytaz, and T. Ebrahimi, "Context-dependent privacy-aware photo sharing based on machine learning," in *IFIP International Conference on ICT Systems Security and Privacy Protection*. Springer, 2017, pp. 93–107.

[12] I. Bilogrevic, K. Huguenin, B. Agir, M. Jadliwala, M. Gazaki, and J.-P. Hubaux, "A machine-learning based approach to privacy-aware information-sharing in mobile social networks," *Pervasive and Mobile Computing*, vol. 25, pp. 125–142, 2016.

[13] Z. Xu and M. Xia, "Distance and similarity measures for hesitant fuzzy sets," *Information Sciences*, vol. 181, no. 11, pp. 2128–2138, 2011.

[14] C. Jiang and P. Steenkiste, "A hybrid location model with a computable location identifier for ubiquitous computing," in *International Conference on Ubiquitous Computing*. Springer, 2002, pp. 246–263.

[15] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1994, pp. 133–138.

[16] F. Delva, A. Edjolo, K. Peres, C. Berr, P. Barberger-Gateau, and J. Dartigues, "Hierarchical structure of the activities of daily living scale in dementia," *The journal of nutrition, health & aging*, vol. 18, no. 7, pp. 698–704, 2014.

[17] T. Li, H. Yang, J. He, and Y. Ai, "A social network analysis methods based on ontology," in *2010 Third International Symposium on Knowledge Acquisition and Modeling*. IEEE, 2010, pp. 258–261.

[18] Z. Alom, Barbara Carminati and E. Ferrari, "Adapting users' privacy preferences in smart environments," in *IEEE International Congress on Internet of Things Services (ICIOT 2019)*.

[19] G. A. Seber and A. J. Lee, *Linear regression analysis*. John Wiley & Sons, 2012, vol. 329.

[20] "mlr: Machine Learning in R ," accessed on April 2019. [Online]. Available: https://rdrr.io/cran/mlr/

[21] M. Gan, X. Dou, and R. Jiang, "From ontology to semantic similarity: calculation of ontology-based semantic similarity," *The Scientific World Journal*, vol. 2013, 2013.