Enhancing User Privacy in Mobile Devices Through Prediction of Privacy Preferences[⋆]

Ricardo Mendes[1] [0000-0003-2077-7223], Mariana Cunha[2] [0000-0002-8920-8128],
João P. Vilela[2] [0000-0001-5805-1351], and
Alastair R. Beresford[3] [0000-0003-0818-6535]

[1] CISUC and Department of Informatics Engineering, University of Coimbra, 3030-290 Coimbra, Portugal
{rscmendes,mccunha}@dei.uc.pt

[2] CRACS/INESCTEC, CISUC and Department of Computer Science, Faculty of Sciences, University of Porto, Portugal
jvilela@fc.up.pt

[3] Computer Laboratory, University of Cambridge, Cambridge, UK
arb33@cam.ac.uk

Abstract. The multitude of applications and security configurations of mobile devices requires automated approaches for effective user privacy protection. Current permission managers, the core mechanism for privacy protection in smartphones, have shown to be ineffective by failing to account for privacy's contextual dependency and personal preferences within context. In this paper we focus on the relation between privacy decisions (e.g. grant or deny a permission request) and their surrounding context, through an analysis of a real world dataset obtained in campaigns with 93 users. We leverage such findings and the collected data to develop methods for automated, personalized and context-aware privacy protection, so as to predict users' preferences with respect to permission requests. Our analysis reveals that while contextual features have some relevance in privacy decisions, the increase in prediction performance of using such features is minimal, since two features alone are capable of capturing a relevant effect of context changes, namely the category of the requesting application and the requested permission. Our methods for prediction of privacy preferences achieved an F1 score of 0.88, while reducing the number of privacy violations by 28% when compared to the standard Android permission manager.

Keywords: Permission Managers · Contextual Integrity · Privacy as Expectations · Mobile Devices · Android.

[⋆] This work is supported by project COP-MODE, that has received funding from the European Union's Horizon 2020 research and innovation programme under the NGI TRUST grant agreement no 825618, and the project SNOB-5G with Nr. 045929 (CENTRO-01-0247-FEDER-045929) supported by the European Regional Development Fund (FEDER), through the Regional Operational Programme of Centre (CENTRO 2020) of the Portugal 2020 framework and FCT under the MIT Portugal Program. Ricardo Mendes and Mariana Cunha wish to acknowledge the Portuguese funding institution FCT - Foundation for Science and Technology for supporting their research under the Ph.D. grant SFRH/BD/128599/2017 and 2020.04714.BD, respectively.

1 Introduction

In the current age of information, the rich and pervasive data collection sparks new applications (apps) that foster advances in our society. In this context, smart and mobile devices are of paramount importance due to their inherent sensory capacity. However, this data exchange often weights on the privacy of each individual, whose practiced trade-off is not often perceived or understood.

To empower users with control over their privacy, smartphones have implemented permission managers (PMs) that control, with user oversight, which resources, such as sensors and data, can be accessed by each application. Under the runtime permission system, the current mechanism employed in both Android and iOS, apps must require user permission the first time they require access to a sensitive resource. When presented with the prompt request, the user may either deny or allow the request for this single time, which will enforce the app to request the next time it needs the same access, or allow indefinitely, an option that can then be changed in the settings of the phone.

The runtime permission system has been positively received by users, who report being more in control over their privacy [3,5]. Its biggest drawback however, lies on the amount of permissions that are allowed without user intervention or even awareness. Specifically, after allowing a permission, the app can generally access the resource at any time and for any purpose even when the user is unaware that the app is running. In this case, the user may deny the permission by going to the phone settings, a practice that is seldomly used [3].

Automatically allowing permissions stems from the necessity to increase usability as apps make hundreds of permission checks per day [2,13]. Asking on every use would be the best theoretical privacy choice, but constant warnings lead users to fatigue and habituation [7], a state where individuals become desensitized and therefore promptly dismiss notices. Undesirably, current PMs automate permission requests without regard for the context, thus violating contextual integrity, that is, incurring in data collection practices that defy the norms and expectations at the given surrounding context [14]. Therefore, the current trade-off between privacy and usability bestowed by the Android PM is insufficient, and in fact, it results in a violation of privacy in 15% of times [13]

In a previous work we have collected and analyzed the expectation of users regarding permission decisions within their surrounding context [13]. Our results showed that the grant result, that is whether the user allows or denies a permission, sees the strongest correlation with user expectation [13]. Moreover, both user expectation and grant result varied with changes in the context. In this paper we analyze this dynamic by measuring the importance of the context in privacy decisions using the same dataset. We then leverage such relation to develop an automated, personalized and context-aware permission model. This paper makes the following contributions:

- We empirically uncover an intrinsic relation between the pair category of the requesting app – requested permission, and user context. This relation advents from the fact that different apps are used under different contexts, therefore conditioning the permission requests that are prompted to the user.

- We develop a personalized automated PM for prediction of privacy decisions by taking into consideration the expectation, user and phone context, thus achieving a ROC AUC of 0.96 and an F1 score of 0.92. Without user expectation, which is the strongest correlated feature with privacy decisions but requires user input which we seek to minimize [13], we achieve a ROC AUC of 0.9 and an F1 score of 0.88.
- Finally, our automated solution is able to reduce the number of privacy violations by 60% when compared to a standard Android handset. Without using the expectation as input feature for the prediction, these violations can still be reduced by 28%.

The remainder of this paper is structured as follows. Section 2 contextualizes the problem by providing related work. Section 3 presents the dataset used in this work and an exploratory data analysis to uncover the relation between privacy decisions and surrounding context. In Section 4 we leverage such relation to train personalized and context-aware models to predict privacy decisions. Section 5 presents some limitations and future work and Section 6 concludes this work.

2 Related Work

With runtime permissions, apps must request permission the first time they require access to a sensitive resource, thus allowing a fine-grained control over each particular permission for any app [5]. By prompting at runtime, permission requests are contextualized by the need of the app at the time of the prompt, therefore helping users to make an informed decision [3,5].

The major problem with the current runtime permission model lies not in the permission prompts, but in the resource accesses that are made without the user knowledge [2,22,23]. After being granted once, apps generally have access to a resource until the user denies it through phone settings, which they typically do not [3] or, in newer Android versions (from Android 11) until it is automatically reverted to the denied state after a few months of not using the app.

The automated management of privacy decisions is made necessary by the number of sensitive resource accesses that apps make – hundreds per day [13]. In fact, users feel their personal space violated when confronted with apps' intrusive practices [2,19]. Regrettably, the automated approach taken in Android runtime PMs incurs in the violation of privacy in over 15% of times [13], meaning it still fails to effectively protect users [5,18].

The design of automated approaches must consider privacy's characteristics, namely, varying individual preferences within each surrounding context [1], i.e. be personalized and context-aware. In this regard, complex contextual modelling techniques have been proposed for policy-based PMs. However, these require expertise to setup that the average user does not have [7,8,18] Therefore, approaches that streamline context modelling to the simple use of contextual features in the prediction have the advantage to facilitate the automation and therefore improve usability. This type of features can describe the state of the phone [24,25] and the state of the user [15].

To capture personalized privacy preferences, one can naively build a prediction model for each user by training with their responses to permission requests. However, this requires a considerable amount of user input [15]. A better approach is to have a classifier boostrapped with data from multiple people and then personalize it iteratively as the user answers a few more requests [24]. Nevertheless, if one starts considering more features for the prediction model, the number of permission requests required from the user, i.e. input, exponentially grows.

A different approach towards personalized privacy is to build and assign privacy profiles, that is, a set of predefined rules that are defined according to user preferences [9, 12]. This line of work showed that while people's privacy preferences are diverse, a small number of privacy profiles can effectively capture the vast majority of users' preferences [12]. Furthermore, these profiles can be assigned through a small number of questions, therefore reducing the amount of required input from users [11].

Our work builds up on previous approaches towards automated privacy enforcement by considering the personalization that is granted by privacy profiles and contextual features to develop an automated, personalized and context-aware PM. However, we differentiate ourselves by considering and evaluating the impact of contextual features and user expectation. By doing so, we depart from the traditional privacy profiles that are built with only the category of requesting app and the requested permission [12], to incorporate context-awareness in the personalization. Towards this goal we analyze a dataset of permission decisions from 93 real users collected in-situ with a particular focus on the relation between privacy decisions and their surrounding context. We then leverage these relations in the development of methods to automatically predict privacy decisions. Finally, we compare the best achieved performance with the default Android PM with respect to the amount of privacy violations, that is, the number of requests that were automatically allowed, but that would have been denied if the user had the opportunity to do so.

3 Permission Decisions in Context

To improve the effectiveness of PMs in protecting user privacy, automation is paramount, which must account for personal preferences within each surrounding context [1]. In this work we first analyze privacy's contextual dependence by evaluating which features are actually relevant towards privacy decisions. We then leverage such relations to build automated, personalized and context-aware models that predict privacy decisions. Towards both goals, we used the COP-MODE dataset [13] whose description we provide in Section 3.1. Section 3.2 provides an exploratory data analysis and respective comparison with existing works. Section 4 then describes the development of the predictive models.

3.1 The Dataset

The COP-MODE dataset [13] is a collection of over 65000 permission decisions and the surrounding context collected under real world conditions. This data was obtained in a set of campaigns spawning from July 2020 up to May 2021 with a total of 93 volunteers. Each campaign consisted of a period of at least one week, where participants would carry a phone, pre-installed with a PM that prompts for user input at every *permission check*. This PM would collect the input and contextual features at the time of the prompt. While the dataset contains more data [13], we focus on the following fields that are of relevance to this work:

- Requesting Application: name, app category from the Play Store and visibility at the time of the request. An app is in the foreground if it either has an activity visible to the user or a service with a foreground notification.
- Permission: name and group of the requested permission and user response.
- Phone state: plug and call states, and network connection type.
- User context: current time, semantic location and whether the user is in an event or not, as returned by their calendar. The semantic location was collected from user input, whose possibilities were “home”, “work”, “travelling” or “other”.
- Expectation: the participant has to answer the question (translated from Portuguese) “For what you were doing with the phone, is this request expected?” with: yes, no or do not know.

We should note that the dataset is biased towards young adults with technical expertise [13]. Therefore, the phone usage and privacy preferences might differ from a more diverse population. However, the methodology towards building predictive models and the achieved performances from Section 4 should apply and endure in general.

3.2 Exploratory Analysis

The dataset contains 2180302 permission requests collected from the 93 participants at an average of 836.85 requests per day and per participant with a standard deviation (std) of 19.15, or 34.87 ($std = 0.8$) per hour. These numbers prove that an ask-on-every-time approach, the ideal privacy choice, is infeasible in practice. Of the total requests, 65261 (2.99%) were answered by participants, corresponding to an average of 25 ($std = 0.42$) answers per day, per participant.

From the 65261 answered requests, participants granted 43263 (66%), while denying the remaining 21998 (33%). To have a holistic view on which permissions are allowed, Figure 6 presents the average grant rate, i.e. the percentage of allowed permissions, per category (y axis) and per permission (x axis), where dark green corresponds to all permissions allowed and dark red to all permissions denied. From the plot we can observe that the majority of categories have grant rates in the interval of $[45, 75]\%$. However some categories present grant rates of over 80% or closer to 0%, but the number of requests from these type of apps is rather small. The exceptions to this observation with a considerable number of requests are the WEATHER category, where 93% of the 370 requests were

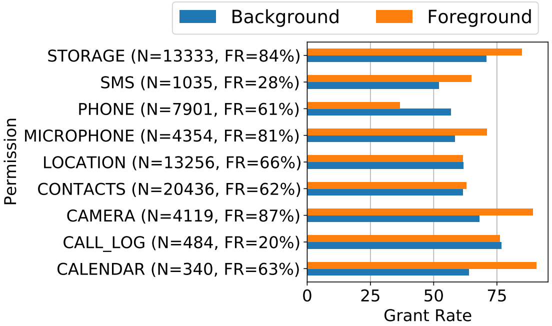


Fig. 1: Grant rate for each permission and whether the requesting app was foreground (visible) or background. The "N" is the number of requests per permission and "FR" the foreground ratio, that is, the percentage of requests that came from apps that were visible to the user at the time of the request.

granted, and GAME (730) and VIDEO_PLAYERS (2413) where almost 80% of requests were denied. It is possible that these latter categories see most of their requests denied because the permissions are not necessary for their primary functionality, which typically leads users towards denying [5]. For instance, some of the requested permissions from apps in the GAME category, such as PHONE, MICROPHONE and CONTACTS are not intuitive with respect to the functionality of this type of apps. As for the grant rate per permission group, the rate is near the interval of $[45, 85]$%. CAMERA, STORAGE and CALENDAR permissions are granted over 80% of the time, which might indicate that when apps request these permissions, there are contextual cues or a clear necessity that lead users to allow.

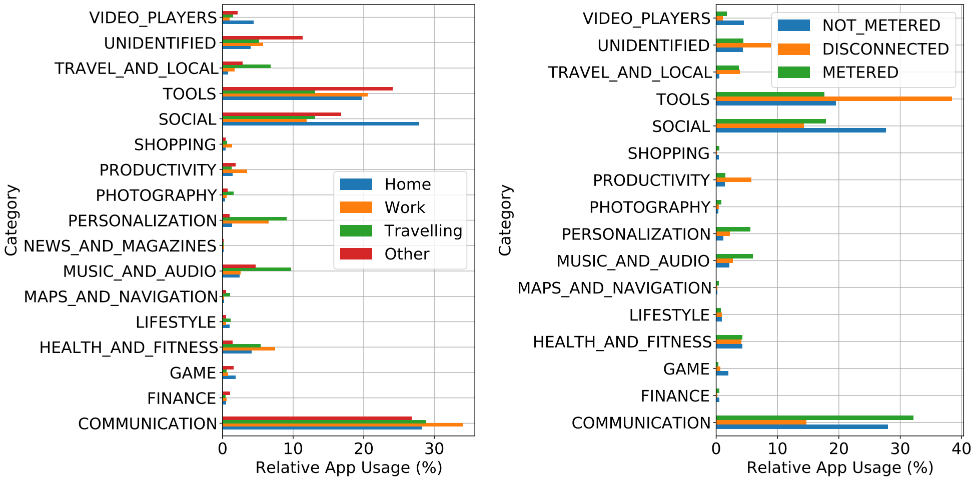
To assess the importance of each feature in the grant rate we measured the information gain of each feature, whose values we leave in Table 2 of Appendix B. The strongest gain advents from user expectation (wasRequestExpected), as analyzed in a previous paper [13]. However, this particular feature requires user input, which we seek to minimize. Unfortunately, we were not able to estimate user expectations with enough accuracy to be then able to use such feature in the prediction of privacy decisions. Thus, we focus on other features towards developing the personalized and context-aware PM. After the expectation, the most important features according to the information gain are some permissions and app categories, the visibility of the requesting app (isRequestingAppVisible), the location of the user (selectedSemanticLoc) and the network status. The following subsections analyze the grant result with respect to each of these latter three contextual features.

Visibility of the Requesting Application Previous work [23] has identified the visibility of the requesting app as one of the most important contextual feature guiding permission decisions. Follow up work from the same authors [20,24] focused on this feature towards predicting the grant result. However, contrary to their conclusions, their feature analysis revealed that the visibility of the app was the feature with the lowest information gain, as can be seen in Appendices A and B of [24]. In our dataset the information gain is almost 8 times higher (c.f. Table 2). However, from the 65261 answered requests, users allowed 68% of requests coming from visible apps and 62% of requests from background apps. This discrepancy is lower than anticipated, which signals that the visibility of the requesting app as a single feature has a low impact in the grant result.

While the overall grant rate between foreground and background requests varies little, this rate can strongly depend on the pairs visibility-category of the requesting app and visibility-requested permission. Due to space constraints we omit the grant rate per visibility and per category, and present only the grant rate for each permission and each visibility of the requesting app in Figure 1. From this plot we observe that CONTACTS, CALL_LOG and LOCATION requests are allowed equally regardless of the visibility, while STORAGE, SMS, MICROPHONE, CAMERA and CALENDAR are more often allowed when requested from the foreground than from the background. Finally, the PHONE permission is the only permission that is more often allowed from the background. We have have no justification for this latter result as a limitation of the dataset is not collecting the reasoning for some privacy choices [13]. Nevertheless, we can conclude that while the visibility of the requesting app alone has low impact on the privacy decision, which contrasts with previous findings [23], the combination with other features such as the permission and category might improve prediction performance. We further examine this correlation in Section 4.

User Location and Network Status According to Table 2, the mutual information gain between the grant result and the user location is high. Looking at the grant rate, users allowed 65% of requests while at home, 85% while travelling, 74% while at work and 57% in other locations. This variance is relevant, specially for when the user is travelling, where they accept almost 9 out of 10 requests. There are two main reasons for the observed variances in the grant rate for each location: privacy preferences vary with the user location [1]; and the app usage also varies with each location, as analyzed next. Other factors can contribute to the discrepancies, such as lack of time to thoughtfully answer prompts when travelling or working, potentially leading users to allow everything. However, these are situational and would require more data to empirically evaluate.

As shown in Figure 6, different app categories have varied grant rates. Therefore, if different apps are used in different locations, it is expected that the grant rate also varies implicitly. Figure 2a presents the relative app usage in percentage given by the apps in the foreground, per semantic location. The relative usage is made per location, such that a fair comparison between locations is achieved, as the dataset is strongly skewed towards the home location. From the plot we can observe that COMMUNICATION, SOCIAL and TOOLS are the most used apps regardless of the location. Additionally, we can clearly see that there are some trends in the type of app usage and the location of the user. Specifically, SOCIAL and VIDEO_PLAYERS apps seem to be predominantly more used at home than in other locations. TRAVEL_AND_LOCAL, PHOTOGRAPHY, PERSON-



(a) Per user location.

(b) Per network status.

Fig. 2: Relative app usage as measured by the relative number of requests in where app from each category were in the foreground. Values inferior to 0.1% were removed from the plot to simplify visualization.

ALIZATION and MUSIC_AND_AUDIO are more used when travelling, which is expected except for the PERSONALIZATION category, while TOOLS are less used when travelling when compared to the other locations, which is also intuitive. Finally, both MAPS_AND_NAVIGATION and LIFESTYLE see a stronger usage when travelling. However, the use of this type of apps was strongly impacted by the COVID19 mobility restrictions, thus presenting a small overall usage. To conclude, the correlation between the location and the grant result can be explained not only because of personal preferences in each location but also due to the types of apps that are used in each context, which, as we have seen in Figure 6, can have diverging grant rates.

Similar conclusions can be made for the network status. From the answered permission requests, 1856 (2%) were captured while the phone was disconnected, 20591 (20%) while connected to a metered network and 80084 (78%) while connected to a non metered network. These numbers indicate that most people are continuously connected to the Internet, although some impact of COVID19 travel restricts can influence this result. The user allows 77% of permission when using a metered network, 64% when using a non-metered network and only 47% when offline. Again, this correlation with the grant result is relevant, as also highlighted by the information gain from Table 2. However, and similarly to user location, the network status is an indication of the context of the user, which in turn influences the apps that are used.

Figure 2b presents the relative app usage per category given by apps in the foreground for each of the network status. We can observe that TOOLS and

		Count	Location Count (%)	Grant Rate (%)
Location	Network Status			
Home	DISCONNECTED	923	1.69	41.93
	METERED	6600	12.11	74.71
	NOT_METERED	46997	86.20	63.61
Other	DISCONNECTED	129	5.81	51.16
	METERED	1273	57.34	59.15
	NOT_METERED	818	36.85	55.87
Travelling	DISCONNECTED	128	3.12	68.75
	METERED	3423	83.39	85.83
	NOT_METERED	554	13.50	83.57
Work	DISCONNECTED	126	2.85	59.52
	METERED	2433	55.10	81.38
	NOT_METERED	1857	42.05	66.34

Table 1: Number, relative count and grant rate of permission requests per semantic location and network status. The grant rate is the percentage of permissions allowed for each pair of location–network status.

PRODUCTIVITY apps are mostly used while offline, while COMMUNICATION and SOCIAL are mostly used online, which is expected. PHOTOGRAPHY, PERSONALIZATION, MUSIC_AND_AUDIO and MAPS_AND_NAVIGATION are mostly used in a metered connection, which as we have seen from Figure 2a, are typically used when travelling. From these observations we conclude that user context, which is partially described by their location and the network status, influences the app usage and therefore the apps that request permissions at these times. In other words, the category of the requested app and the required permission encapsulate contextual information that, while potentially insufficient to describe user context, give clues about the state of the user.

It should be noted that either location, network status or even both are insufficient to effectively describe the variance in the grant rate. For instance, within a single location, the grant rate varies for each network status and vice-versa. Table 1 presents these values for each pair of location-network status. The first observable result from this table is that the location of the user and the network status are strongly correlated. Looking at the "Location Count (%)": when the user is at home, unmetered connections are used over 86% of times; when travelling, metered connections are used 83% of times. At work and other locations, the connection status is more balanced between metered and unmetered connections. However, these latter ratios might vary greatly with each individual. Finally, while some previously mentioned trends endure, the grant rate strongly varies for each pair of location-network status. For instance, the highest and lowest grant rate in any location is when the user is using metered connections and disconnected, respectively. However, under metered networks for instance, if the user is travelling, over 86% of requests are allowed, but if the user is at a location other than the specified three, the grant rate lowers to 59%. These observations allow us to conclude that while location and network status are related, both give contextual cues, even if in the form of the apps that are used in such contexts.

Comparison of the Analysis with Previous Work Earlier studies evaluated the degree of user comfort regarding apps' intrusive data collecting practices [2, 4, 19], which included confronting users with the frequency of access to sensitive resources, relating to the number of permission requests in our work. Many of these results however, were conducted under the older install-time PM.

The runtime PM brought fine-grained permissions and in-context prompts, therefore being positively received by users, who reported being more in control of their privacy [3, 5]. However, these studies only evaluated the permission decisions made at the permission prompts. Regrettably, after being granted once, apps generally have access to a resource until the user denies it through phone settings. The context and the purpose of these automatically granted permissions can greatly vary from the ones at which they were first requested.

Closely related is the work of Wijesekera et al. [23, 24] that evaluates the importance of both contextual and behavioral features on permission decisions. While they use a subset of the current Android permissions, some of the findings coincide and others contrast. Particularly, the percentage of denied permissions and the number of privacy violations are similar, while in opposition, the visibility of the requesting app had low impact in privacy decisions with our dataset. However, contrary to our data collection, their data relates to reported behavior collected after the data practice, which might not align with real behavior. In turn, our data collection tool was also a PM that actually denied apps permissions, and thus incurred in the corresponding usability loss. Furthermore, in addition to analyzing contextual features under the information gain or their contribution to the performance of the classifier [15,23,24], we expand such analysis by exploring the intrinsic relation between different contextual features. As a result, we uncover a relation between the permission prompts that are issued to the user and their context.

4 Predicting Privacy Decisions

The previous section evidenced how privacy decisions vary with changes in the context, and how features and their correlation can discriminate the grant result. In this section, we leverage these relations towards developing and automated, personalized and context-aware PM that predicts these decisions. For fair comparison, we follow a similar methodology to previous works. Specifically, we consider machine learning approaches to automate privacy decisions, while combining privacy profiles [11], context-awareness [15, 25] and user expectation.

To train the classifier for grant result prediction we perform one-hot encoding of the categorical features, such as the requesting app category, and normalize all collected data. We then start by analyzing the performance of a global predictor in Section 4.1 which uses the input features to output the decision to allow or deny a request, while treating each user equally, i.e. without personalization. This evaluation is performed by first selecting the best predictor (model) and respective parameters through a cross-validated grid-search, followed by an evaluation of the best feature set to use in the prediction. We resort to the F1 score metric to compare the performance with previous works and to the Area Under the Receiving Operation Curve (ROC AUC) as performance indicator, as the F1 score presented some misleading results as detailed in the referred section. The global predictor is then used as baseline comparison to the personalized predictors in Section 4.2 where we resort to the use of the privacy profiles as an additional feature in the prediction, for personalization. However, one can consider different feature sets for the creation of profiles and for predicting of the grant result with the profiles. Therefore, to evaluate the combination that leads to the best performance, Section 4.2 presents the results for all considered combinations of feature sets for the creation of the profiles and all considered feature sets for the prediction. All considered feature sets were based on their importance in the grant result as measured by the information gain from Table 2 and from the analysis in the previous section. Finally, Section 4.2 further presents the privacy violations incurred by the best predictors, while contrasting them with the violation rate achieved by the Android PM with our dataset.

4.1 Global Prediction

Since there is no a priori best classifier to predict privacy decisions, we experimented using a grid-search with models from the literature. Specifically, Support Vector Machines (SVM) with linear [11,15,25] and Radial Basis Function (RBF) kernels, decision trees [15], bagging, ada boosting, random forest and a neural network. Although the results for each model were similar, we picked the best performance: ada boost with approximate ROC AUC of 0.827 and F1 score of 0.808. These results were achieved using 100 decision trees with a max depth of 1 as base classifiers and with a learning rate of 0.5, which we use for the remainder of the experiments. We also focus on the ROC AUC, as the F1 score was misleading. Specifically, using the mode prediction model resulted in an F1 score of 0.8 (close to the best performance) but in a ROC AUC of 0.5, which is the same value as a random classifier would achieve.

A 5-fold cross-validated feature forward selection by the ROC AUC selects the expectation as the most important feature, followed by some permissions and categories. The visibility of the requesting app is selected as the seventh most important feature. However, the visibility is highly correlated with the expectation, as previously discussed, and thus, this cumulative forward approach fails to account for individual feature importance. To better evaluate the importance of features, we have considered some feature set variants based on the analysis provided in Section 3.2 and cross-validated the performance of the classifier with each variant. Figure 3 presents the obtained performances, in where it is clear that the expectation is the most relevant feature. In fact, just using the expectation results in an F1 score and ROC AUC of over 0.8. Adding the category and permission to the expectation, leads to the best ROC AUC (≈ 0.831), even slightly better than when using all features. Contextual features such as the [V]isibility, [L]ocation and [N]etwork status added very little or nothing to the category and permission (CP), as can be seen from the similarity of scores between using CP or any combination of V, L and N with CP. These results

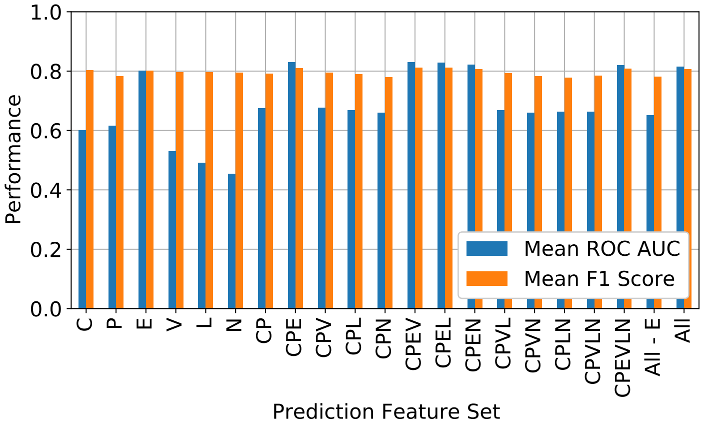


Fig. 3: 5-fold cross-validated performance of the ada boost classifier on the different considered dataset variants. Each variant is a combination of the following features, which are identified by their first letter: [E]xpectation, [C]ategory and [V]isibility of the requesting app, [P]ermission requested, [L]ocation, and [N]etwork status. "All" corresponds to using all features available in the dataset and "All - E" is all features except the expectation.

indicate a general lack of importance of the considered contextual features in the performance of the classifier. However, we believe that at least in part, this is due to the fact that the category of the requesting app and requested permission already encode part of the context as discussed in Section 3.2. Therefore, the additional information gain added by the contextual features is either not sufficient, or the classifier fails to account for it. Regardless, a ROC AUC of over 0.8 is already a good performance for a classifier that treats all users equally, that is, it fails to account for privacy's personal preferences. The next section enhances this approach by providing context-aware personalization.

4.2 Personalized Prediction

Traditionally, privacy profiles are build by applying hierarchical clustering to users [10, 11], where each user is represented as a tensor where each cell is the tendency to allow or deny requests for a particular pair of category-permission (CP). However, our dataset contains additional features that capture the similarity between user behavior in a more fine-grained way. Specifically, instead of just using the pairs of CP, we can additionally consider the [E]xpectation or other contextual features such as the user [L]ocation, the [V]isibility of the requesting app and the [N]etwork status to form context-aware privacy profiles. Towards this end we consider the following feature variants for clustering: CP, CPV, CPE, CPL, CPN, CPVLN and CPEVLN. Furthermore, regardless of how the profiles are formed, we can use any combination of features in the prediction alongside

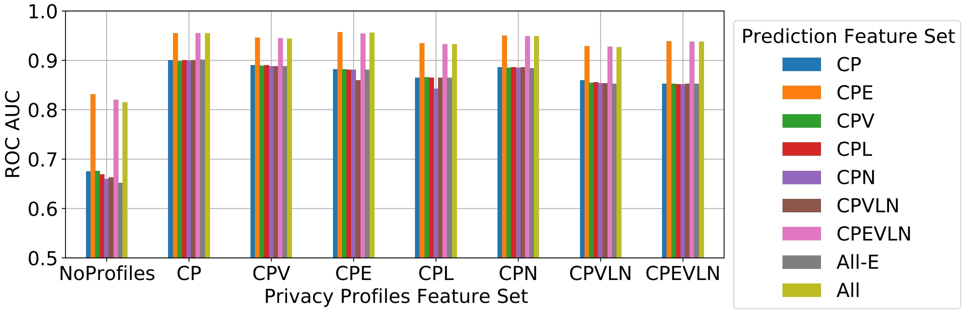


Fig. 4: 5-fold cross-validated performance with privacy profiles built with different feature sets, or no privacy profiles ("NoProfiles"), followed by prediction with several other feature sets. The number of profiles was varied from 1 to 9 and only the best result is displayed for each combination of inputs. Each feature set is identified by the combination of the following features identified by their first capitalized letter: [C]ategory, [P]ermission, [E]xpectation, [V]isibility, [L]ocation, and [N]etwork status. "All" and "All - E" corresponds respectively to using all features and all features except user expectation.

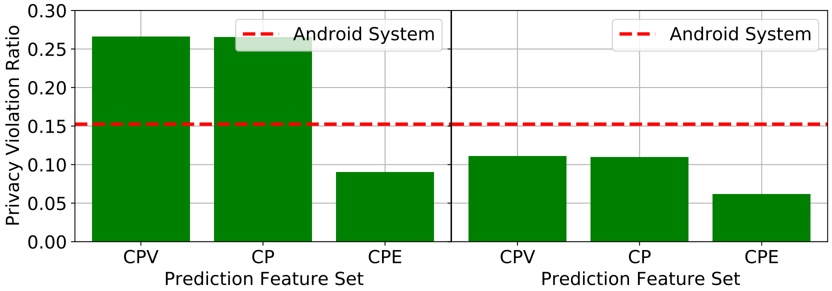
the profiles. Therefore, we performed all combinations of clustering with the feature variants displayed above, with the same feature variants in the predictions plus "All" features and all features except the expectation ("All-E"). For each combination of profiling and prediction, the number of profiles was varied from 1 to 9 and only the best results are displayed.

Figure 4 presents the obtained results, where the first observation is that any profiling with any prediction approach outperforms not using profiles, thus confirming previous findings that personalization improves performance [11,24]. Secondly, the best overall results are achieved by profiling only with CP. This is partially due to the fact that using more features in the profiling increases the amount of missing data that needs to be inputed, therefore potentially biasing the data. Nevertheless, profiling with CPE followed by prediction with all features achieves a ROC AUC of 0.956 or prediction with CPE achieves a ROC AUC of 0.957, where this latter is the best performance. Similar results are achieved by profiling with CP and predicting only with CPE, a ROC AUC of 0.955, approximately. The advantage of this second best result is that less data is required, specially for assigning the privacy profiles, a step that requires asking questions to the user and therefore, should be minimized [11]. Finally, without the expectation, the best performance is achieved by clustering and predicting with CP, a ROC AUC of approximately 0.9.

The previous results are comparable to the state of the art [11], whose reported F1 score was 0.900 with profiles built with the tuples <category, permission, purpose>. Our best F1 score is approximately 0.924, achieved through profiling and predicting with CPE, that is, with the expectation instead of the purpose. Without the expectation, our best F1 score is approximately 0.88, with profiles using only the pair category-permission and predicting with the category, permission, visibility, semantic location and network status (CPVLN). However, because the datasets are different, we cannot say that taking into consideration the expectation results in a better performance than using the purpose. A natural departure from this work is to combine both features.

An interesting, yet unexpected result that is also observable from Figure 4 is the rather low impact of the contextual features in the prediction. Specifically, if the expectation is not considered, using just the category and permission often results in the best performance.This is partially explained by the correlation between the context of the user and the pair category-permission, as discussed in Section 3.2. However, we were expecting a stronger influence, particularly the visibility of the requesting app, which has been found to have a strong influence in privacy decisions [23]. The reason for the low impact of the visibility of the requesting app is that users allow 68.24% of visible requests and 61.87% of background requests, as aforementioned. This difference might be irrelevant to the classifier. A potential reason for the low impact of the location is the fact that 83.54% of requests were with users at home, owed to COVID19 travel restrictions that were in place at the time of the campaigns [13]. Due to this skewness, the importance of the location might be mis-measured. Therefore, we repeated the previous methodology while subsampling the home requests to equal the number of work requests. The results with the subsampled data, whose plots we omit due to space constraints, showed that without profiling, the location feature slightly increased the performance, but with profiling the results were similar to the ones obtained in Figure 4. It is possible that these contextual features, specially the visibility, have a varying importance depending on the user as some users allow/deny everything regardless of any feature, while others are more selective. However, profiling with these features either failed to capture these personal preferences or the increase in the missing data deteriorated the results, due to the increasing amount of missing data. Towards validating the potential bias introduced by the inputted data, we build privacy profiles using the K-means clustering algorithm [16, 17] instead of hierarchical clustering. The performances were worse in all cases, and thus, we omit such results.

Finally, we can compare the number of privacy violations that these approaches incur. Privacy violations are defined as permission requests that the user explicitly denied, but would otherwise be granted. As previously mentioned, for the collected dataset, the Android default PM would have violated the privacy in 15.25% of requests and would have incurred a median of 64 prompts to the user in a period of approximately a week. A personalized and automated prediction following the methodology above would require only a few questions to assign the profile [11] and it would result in 6.18% of privacy violations, a 59.5% reduction on Android PM, as displayed in Figure 5b, where the green bars present the violation ratio for the best personalized predictors and the dashed red line is the Android system violation ratio. Without the expectation, the lowest privacy violation ratio achieved is 11% when predicting with CP, which is still a reduction of 27.9% when compared to the standard Android PM. Looking



(a) Global Prediction (no profiles).

(b) Profiles using CP.

Fig. 5: 5-fold cross-validated privacy violation ratio of the best performant predictors for the global predictors 5a and the personalized predictors 5b. Each feature set is identified by the combination of the following features identified by their first capitalized letter: [C]ategory, [P]ermission, [E]xpectation and [V]isibility, The ratio of privacy violations that the Android PM would have incurred is presented as the red dashed horizontal line.

at Figure 5a, it is noteworthy that automated solutions without privacy profiles, corresponding to the global predictors from Section 4.1, and without expectation, result in a higher amount of privacy violations than the Android system.

In summary, it is possible to automate privacy decisions with high performance, specially when taking into consideration user expectation. Contextual features seem to have a low impact in the performance of the prediction, which we mostly attribute to the fact that the pair category-permission already partially encode the context. Furthermore, the achieved prediction model can reduce the privacy violations in over 50% when compared to the current Android permission system. However, such system requires knowing the expectation of the user regarding every request, which we were unable to predict with sufficient accuracy and would therefore require user input, that should optimally be minimized. Without the expectation, it is possible to automate privacy decisions, while reducing the privacy violations by 27.9%. These results indicate that permission systems can still be enhanced, specially by taking the expectation of users into account.

5 Limitations and Future Work

As referred in Section 3.1, the dataset considered in this work is biased towards young adults with technical expertise. This occurrence advents from the fact that the data was collected whilst COVID-19 restrictions were enforced [13], and we relied on students with on-site classes. We leave for future work to conduct a campaign with a more diverse population as to better validate our findings.

One of the disadvantages of incorporating the expectation in the automation of privacy decisions is that it requires input from the user. We attempted to predict the expectation following a similar methodology to the one described for the privacy decision, including profiling. However, the performance was not high enough to increase the results displayed in Figure 4. This is an indicator that the expectation can be more personal and dynamic than the respective privacy decisions. As future work we intend to analyze possible venues towards improving the prediction of expectations.

An underlying limitation of all automated privacy decision systems, including the approach described in this work, lies on the legal basis of such decisions. Specifically, an automated response to a permission request might not constitute legal consent. Regulations such as the EU General Data Protection Regulation (GDPR) mandate express and unambiguous consent from the user before collecting any personal data [21]. Unfortunately, the GDPR does not provide guidelines for automating privacy decisions. Therefore, further legal discussion will be required. In the meantime, the personalized prediction of privacy decisions can be instead offered as recommendations to the user [11], instead of fully automation. Such approach can mitigate potential challenges of configuring complex privacy systems, such as the lack of expertise by the average user [18].

Finally, a natural departure of this work would be to assess the feasibility of assigning the privacy profiles, the performance of the predictions, and the perceived usability of such PM. Such endeavour requires a new field study. Moreover, the dependence on user data for clustering users and predicting their privacy decisions is a drawback of this approach. To address this issue, we have proposed a clustering mechanism and federated prediction approach [6] with privacy guarantees.

6 Conclusion

The complexity of mobile devices require automation for the management of user privacy. However, the current approach, i.e. the runtime permission model, often violates user privacy, thus failing at protecting the user. The root of this ineffectiveness advents from the non consideration of contextual dynamism and personal preferences within each context that are natural factors impacting privacy decisions. In this paper we analyze a dataset of privacy decisions and their surrounding context to uncover their intrinsic relation. Our analysis reveals that the visibility of the requesting app, the location of the user and the network status are important contextual cues that partially explain the variability of the grant result, i.e., the user decision to allow or deny a permission. In addition, we find that the category of the requesting app and the requested permission moderately encode the context, as the user uses different apps under different contexts. We then leverage such analysis to train models towards building an automated, personalized and context-aware permission manager for prediction of the grant result. Our results show that by taking into account the expectation of the user, one can reduce the number of privacy violations by over 50% when compared to the Android permission manager. Without user expectation, it is still possible to reduce the privacy violations by approximately 28%.

References

1. Acquisti, A., Brandimarte, L., Loewenstein, G.: Privacy and human behavior in the age of information. Science **347**(6221), 509–515 (2015)
2. Almuhimedi, H., Schaub, F., Sadeh, N., Adjerid, I., Acquisti, A., Gluck, J., Cranor, L.F., Agarwal, Y.: Your location has been shared 5,398 times!: A field study on mobile app privacy nudging. In: Proceedings of the 33rd annual ACM conference on human factors in computing systems. pp. 787–796. ACM (2015)
3. Andriotis, P., Stringhini, G., Sasse, M.A.: Studying users' adaptation to Android's run-time fine-grained access control system. Journal of Information Security and Applications **40**, 31–43 (2018)
4. Balebako, R., Jung, J., Lu, W., Cranor, L.F., Nguyen, C.: "Little brothers watching you": raising awareness of data leaks on smartphones. Proceedings of the Ninth Symposium on Usable Privacy and Security - SOUPS '13 p. 1 (2013)
5. Bonné, B., Peddinti, S.T., Bilogrevic, I., Taft, N.: Exploring decision making with android's runtime permission dialogs using in-context surveys. In: Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017). pp. 195–210. USENIX Association, Santa Clara, CA (2017)
6. Brandão, A., Mendes, R., Vilela, J.P.: Prediction of mobile app privacy preferences with user profiles via federated learning. In: Proceedings of the Twelveth ACM Conference on Data and Application Security and Privacy. pp. 89–100 (2022)
7. Felt, A.P., Ha, E., Egelman, S., Haney, A., Chin, E., Wagner, D.: Android permissions: User attention, comprehension, and behavior. In: Proceedings of the Eighth Symposium on Usable Privacy and Security (SOUPS 2012). Association for Computing Machinery, New York, NY, USA (2012)
8. Kelley, P.G., Consolvo, S., Cranor, L.F., Jung, J., Sadeh, N., Wetherall, D.: A conundrum of permissions: installing applications on an android smartphone. In: International Conference on Financial Cryptography and Data Security. pp. 68–79. Springer (2012)
9. Lin, J., Liu, B., Sadeh, N., Hong, J.I.: Modeling users' mobile app privacy preferences: Restoring usability in a sea of permission settings. In: 10th Symposium On Usable Privacy and Security (SOUPS 2014). pp. 199–212. USENIX Association, Menlo Park, CA (2014)
10. Lin, J., Liu, B., Sadeh, N., Hong, J.I.: Modeling Users' mobile app privacy preferences: Restoring usability in a sea of permission settings. In: 10th Symposium On Usable Privacy and Security (SOUPS 2014). pp. 199–212. USENIX Association, Menlo Park, CA (Jul 2014)
11. Liu, B., Andersen, M.S., Schaub, F., Almuhimedi, H., Zhang, S., Sadeh, N., Acquisti, A., Agarwal, Y.: Follow my recommendations: A personalized privacy assistant for mobile app permissions. In: Symposium on Usable Privacy and Security (2016)
12. Liu, B., Lin, J., Sadeh, N.: Reconciling mobile app privacy and usability on smartphones: could user privacy profiles help? In: Proceedings of the 23rd international conference on World wide web - WWW '14. pp. 201–212 (2014)
13. Mendes, R., Brandão, A., Vilela, J.P., Beresford, A.R.: Effect of user expectation on mobile app privacy: A field study. In: 2022 IEEE International Conference on Pervasive Computing and Communications (PerCom). pp. 207–214 (2022)
14. Nissenbaum, H.: Privacy as contextual integrity. Wash. L. Rev. **79**, 119 (2004)
15. Olejnik, K., Dacosta, I., Machado, J.S., Huguenin, K., Khan, M.E., Hubaux, J.P.: Smarper: Context-aware and automatic runtime-permissions for mobile devices.

In: 2017 IEEE Symposium on Security and Privacy (SP). pp. 1058–1076. IEEE (2017)

16. Ravichandran, R., Benisch, M., Kelley, P.G., Sadeh, N.M.: Capturing social networking privacy preferences. In: International symposium on privacy enhancing technologies symposium. pp. 1–18. Springer (2009)
17. Sanchez, O.R., Torre, I., He, Y., Knijnenburg, B.P.: A recommendation approach for user privacy preferences in the fitness domain. User Modeling and User-Adapted Interaction **30**(3), 513–565 (2020)
18. Shen, B., Wei, L., Xiang, C., Wu, Y., Shen, M., Zhou, Y., Jin, X.: Can systems explain permissions better? understanding users' misperceptions under smartphone runtime permission model. In: 30th USENIX Security Symposium (USENIX Security 21) (2021)
19. Shklovski, I., Mainwaring, S.D., Skúladóttir, H.H., Borgthorsson, H.: Leakiness and creepiness in app space: Perceptions of privacy and mobile app use. In: Proceedings of the 32nd annual ACM conference on Human factors in computing systems. pp. 2347–2356. ACM (2014)
20. Tsai, L., Wijesekera, P., Reardon, J., Reyes, I., Egelman, S., Wagner, D., Good, N., Chen, J.w., Clara, S.: Turtle Guard: Helping Android Users Apply Contextual Privacy Preferences. Symposium on Usable Privacy and Security (SOUPS) 2017 (Soups) (2017)
21. Union, E.: Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation). Official Journal L110 **59**, 1–88 (2016-05-04)
22. Votipka, D., Rabin, S.M., Micinski, K., Gilray, T., Mazurek, M.L., Foster, J.S.: User comfort with android background resource accesses in different contexts. In: Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018). pp. 235–250 (2018)
23. Wijesekera, P., Baokar, A., Hosseini, A., Egelman, S., Wagner, D., Beznosov, K.: Android permissions remystified: A field study on contextual integrity. In: USENIX Security. vol. 15 (2015)
24. Wijesekera, P., Baokar, A., Tsai, L., Reardon, J., Egelman, S., Wagner, D., Beznosov, K.: The Feasibility of Dynamically Granted Permissions: Aligning Mobile Privacy with User Preferences. In: Proceedings - IEEE Symposium on Security and Privacy. pp. 1077–1093 (2017)
25. Wijesekera, P., Reardon, J., Reyes, I., Tsai, L., Chen, J.W., Good, N., Wagner, D., Beznosov, K., Egelman, S.: Contextualizing privacy decisions for better prediction (and protection). In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. pp. 268:1–268:13. CHI '18, ACM, New York, NY, USA (2018)

A Grant Rate

Figure 6 presents the average grant result for each pair of category of the requesting app and requested permission.

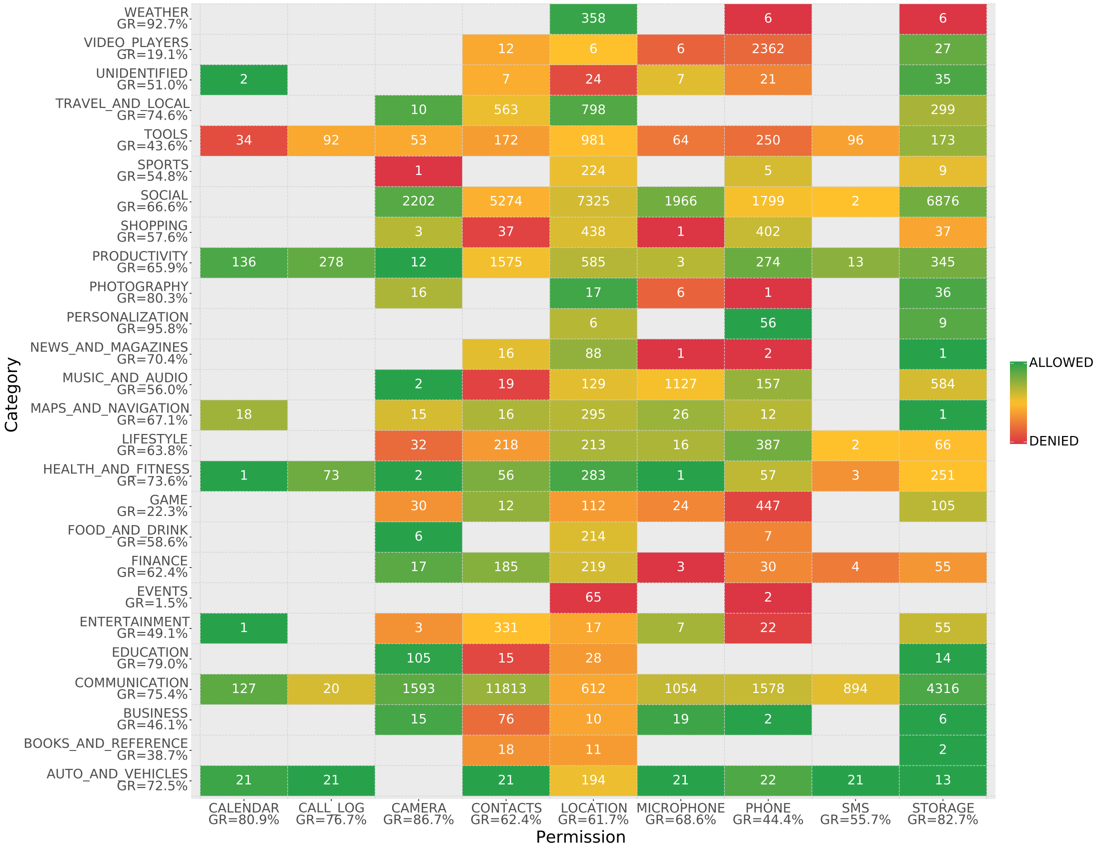


Fig. 6: Average grant result for each pair of category-permission. The number in each cell is the number of requests for the respective pair category-permission group, and GR is the grant rate for the respective category or permission. Categories and permissions with less than 10 requests were removed.

B Information Gain

Table 2 presents the information gain for the grant result with each other feature in the dataset, where categorical features were one-hot encoded.

Table 2: Information Gain for the grant result with every other feature. Showing only values greater than 0.

	grantResult
wasRequestExpected	0.182551
permission_STORAGE	0.018125
category_VIDEO_PLAYERS	0.016729
category_COMMUNICATION	0.013858
permission_PHONE	0.013843
selectedSemanticLoc_Home	0.011375
networkStatus_METERED	0.008086
isRequestingAppVisible	0.007845
networkStatus_NOT_METERED	0.007624
permission_CAMERA	0.006610
permission_CONTACTS	0.005629
selectedSemanticLoc_Travelling	0.005558
category_GAME	0.005136
category_TRAVEL_AND_LOCAL	0.004675
plugState	0.003084
category_WEATHER	0.002793
isTopAppRequestingApp	0.002483
category_TOOLS	0.002355
category_MUSIC_AND_AUDIO	0.002348
permission_LOCATION	0.002234
hour	0.002170
permission_CALL_LOG	0.001951
category_PERSONALIZATION	0.001940
selectedSemanticLoc_Work	0.001710
permission_SENSORS	0.001467
category_BUSINESS	0.001452
category_SOCIAL	0.001350
category_SPORTS	0.000973
category_SHOPPING	0.000963
category_HEALTH_AND_FITNESS	0.000801
isWeekend	0.000623
category_MEDICAL	0.000485
callState	0.000389
category_LIFESTYLE	0.000265
category_ENTERTAINMENT	0.000156
category_FOOD_AND_DRINK	0.000122
networkStatus_DISCONNECTED	0.000103