A Preliminary Attempt of an Intelligent System Predicting Users' Correctness of Notifications' Sender Speculation

Tang-Jie Chang
National Chiao Tung University
Hsinchu, Taiwan
tjchang.cs08g@nctu.edu.tw

Jian-Hua Jiang Chen
National Chiao Tung University
Hsinchu, Taiwan
novel860329.cs08g@nctu.edu.tw

Hao-Ping Lee
National Chiao Tung University
Hsinchu, Taiwan
dimension4.cs03@nctu.edu.tw

Yung-Ju Chang
National Chiao Tung University
Hsinchu, Taiwan
armuro@nctu.edu.tw

ABSTRACT

Prior interruptibility research has focused on identifying interruptible or opportune moments for users to handle notifications. Yet, users may not want to attend to all notifications even at these moments. Research has shown that users' current practices for selective attendance are through speculating about notification sources. Yet, sometimes the above information is insufficient, making speculations difficult. This paper describes the first research attempt to examine how well a machine learning model can predict the moments when users would incorrectly speculate the sender of a notification. We built a machine learning model that can achieve an recall: 84.39%, precision: 56.78%, and F1-score of 0.68. We also show that important features for predicting these moments.

KEYWORDS

Notification source; receptivity; intelligent system

ACM Reference Format:

Tang-Jie Chang, Jian-Hua Jiang Chen, Hao-Ping Lee, and Yung-Ju Chang. 2020. A Preliminary Attempt of an Intelligent System Predicting Users' Correctness of Notifications' Sender Speculation. In *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers (UbiComp/ISWC '20 Adjunct), September 12–16, 2020, Virtual Event, Mexico.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3410530.3414390

1 INTRODUCTION

Mobile phones have become an indispensable part of our daily life. Users receive numerous notifications on the phone every day. However, they tend to selectively view notifications that are interesting or important to them and ignore or dismiss notifications otherwise [2, 6]. In particular, for those notifications related to communication, prior research has shown that users are more attentive to the notifications from the people they are closer with [3, 4]. Furthermore, Chang et al. showed that users' such selective attention, or preference, toward notifications from specific sources often has manifested since the users notice the arrival of the notifications. This is observed from their attendance to only certain notifications after they have speculated about who is likely to send these notifications and then determine whether to attend to them [1]. Furthermore, Chang et al. showed that correct speculation of the sender of the notifications allows users to effectively and selectively attend to notifications they truly want to read and save time from reading those they think unnecessary to read at the moment. Nevertheless, such speculations can be difficult at certain moments such as when users may simply have no clues about who may send the current notification, or when they can associate the notifications with many possible senders. However, currently there has not been research attempt aimed to assist users at these moments, i.e. helping them more likely to accurately judge notification senders to facilitate their selective attendance. Yet, it is also important to recognize that to provide such assistance, the first step would be accurately recognized when these challenging moments take place.

In this paper, we take the first step that explores the feasibility of predicting the moments when users would incorrectly judge the sender of the incoming notification. In addition, through such exploration, we aim to identify features that are predictive of these moments. Note that we focus on speculations about the sender because it has been shown that users can generally accurately tell the applications of incoming notifications based on the alert, whereas their speculation about the sender is much less often accurate [1].

Using the dataset collected from [1], which consisted of users' speculation behaviors of 667 notifications, we have tried four models including Adaboost, Support Vector Machine(SVM), Random Forest, and XGBoost, respectively. We examined their performances in predicting the moments when the users' speculations about the sender were incorrect. Among the four models, Random Forest achieved the highest F1-score (0.676), with recall: 84.39%, and precision: 56.78%. In addition, we show the importance of the top features in terms of their contribution to the prediction.

UbiComp/ISWC '20 Adjunct, September 12–16, 2020, Virtual Event, Mexico

ACM ISBN 978-1-4503-8076-8/20/09.
https://doi.org/10.1145/3410530.3414390

2 RESULTS

2.1 Training-data Collection

This paper used phone data which were collected from previous work [1]. The dataset contained 34 users reporting a total of 1,869

Table 1: Types of Recent Actions We Used

Recent action	Discription
UseSamePackageApp	User use same package app
UseSameMsgPackageApp	User use same message package app
UseSameMsgCateApp	User use same message package category app (e.g. Messenger and Line all categorize to 'messenger')
UseMsgApp	User use message app
ExistScreenOn	User's mobile phone exist screen on
ExistAction	User's mobile phone exist action with the screen
ExistNoti	User's mobile phone exist notification
ExistMessageNoti	User's mobile phone exist message's notification
ExistSameCateNoti	User's mobile phone exist same category's notification
ExistSamePackageNoti	User's mobile phone exist same package's notification

Table 2: Feature category and example

Feature Category	Example
User actions	UseSamePackageAppMinute1, ExistScreenOnInMinute5, UseSameMsgCateAppMinute10, ExistActionInMinute30, UseMsgAppMinute60
Notification activity	ExistMessageNotiInMinute1, ExistNotiInMinute5, ExistSameCateNotiInMinute10, IsMore(text), ExistSamePackageNotiInMinute30
Phone context	DayOfWeek, IsVibrate, IsIndoor, IsMoving, proximity

instances whether they reported to correctly or incorrectly speculate about the senders of the incoming notification. Each instance contained the information about the incoming notification, a label of speculation (i.e. "not able to tell", "incorrect", and "correct"), and the phone contextual information at the moment, including location, activity, phone sensor data, and phone status (e.g., network, battery level, and charging state), and users' actions. User-action information was logged via the Android Accessibility Service [1]. The speculation labels were obtained via experience sampling method (ESM) questionnaires that were triggered by a research app. Specifically, each ESM questionnaire asked users to report their experience with three sampled notifications, including whether they had 1) seen these notifications, 2) sensed them, 3) speculated about their sources, 4) attended to them, and 5) considered their attendance decisions to have been helpful or not. The ESM questionnaires were sent out 6-10 times per day, beginning at least 30 minutes after the participant had started using his or her phone; and the three notifications they asked about were selected at random from among all those that had arrived within the previous 30 minutes.

Among the 1,869 instances, the participants missed 47.6% of these 1,869 notification events. Among the 52.4% of the notifications that they did sense, they reported having made speculations about their sources 71.6% (667) of the time.

2.2 Model Construction

We built model predicting when users would not be able to speculate the notification sender correctly, which included those they self-reported as *not able to tell the source* and those they made incorrect speculations. These instances were assigned a *incorrect*label. The others which users could correctly judge the senders were assigned a *correct* label.

Feature Extraction. We initially extracted 75 features from the collected contextual data. Note that some of the data collected from the phone regarded user-operable features such as ringer mode and phone charging state, while other data such as recent actions (showed in Table 1) had to be applied with feature transformation to provide high-level meanings. And we categorized 75 features into three groups: 1) user actions (32 features); 2) notification activity (22 features) and 3) phone context (21 features). Table 2 list five examples for each three groups. Specifically, we measured whether a research-relevant event had taken place, or a research-relevant condition had prevailed, within periods of 1, 5, 10, 30, and 60 minutes prior to each notification's arrival: for example, *ExistNotiInMinute5* means whether the user had received a notification in the last 5 minutes, *UseMsgAppMinute60* means whether the user had used a messaging app in the last hour, and so forth. We converted categorical variables into numerical values by using a one-hot encoding strategy, e.g., converting ringer mode into the two binary variables *IsNormal* and *IsVibrate*. We eliminated notification events from which any of the selected features were missing and left a total of 439 data for speculation of notifications' sources. Among the 439 responses, we got 227 and 212 for speculated sender correctly and incorrectly, respectively.

Model Choice. We assessed the model performance across four classifiers: 1) Adaboost; 2) support vector machine (SVM), with linear kernel; 3) XGBoost; and 4) Random Forest. Since we had no hypotheses about which time windows or features would boost or impede model performance, we compared the classification results

[1]https://developer.android.com/reference/android/accessibilityservice/AccessibilityService

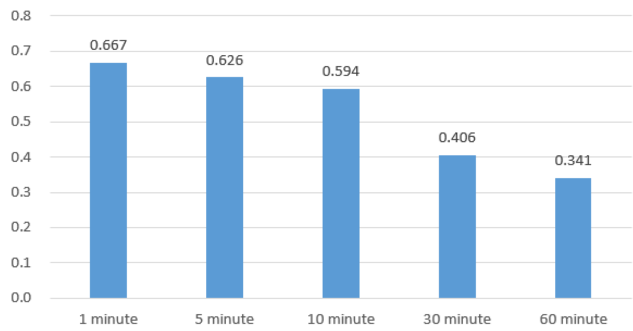

Figure 1: Average F1-score for each time window.

across various combinations of features and time windows. F1-scores were used to make comparisons among classifiers because they account for both precision and recall. Firstly, we select the time window that achieved the best prediction performance. As shown in our results (see Figure 1), the time window of '1 minute' yielded the best averaged F1-score (0.667), with the 5, 10, 30 and 60 minutes alternative yielded 0.626, 0.594, 0.406 and 0.341, respectively. Because the three time windows 'within 10 minutes' outperformed those 'more than 10 minutes' time windows, our analysis later focused on time windows of '1 minute', '5 minute', and '10 minute'.

Using these time windows, our results show that Random Forest outperformed the other classifiers among all the time windows. It achieved an averaged F1-score of 0.538 (see Figure 2) among the three time windows, whereas SVM, Adaboost and XGBoost achieved 0.523, 0.524 and 0.522, respectively. In later sections, we report our results using the model built with Random Forest.

Feature Selection. We built the model with the three time windows ('1 minute', '5 minute', and '10 minute') that achieved the best performance using Random Forest (see Figure 3). Our objective was to identify features predictive of the moments at which users could not correctly speculate the sender. After the feature selection process (see Table 3), our model achieved an recall: 84.39%, precision: 56.78%, and F1-score of 0.676 with 'within 1 minute' features. This implies that the model could capture most of the moments when the users could not correctly judge the sender of the incoming notification. But the detection can contain a considerable portion of false positive.

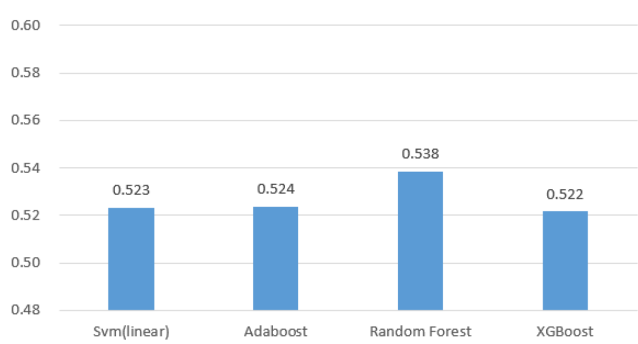

Figure 2: Average F1-score for each model.

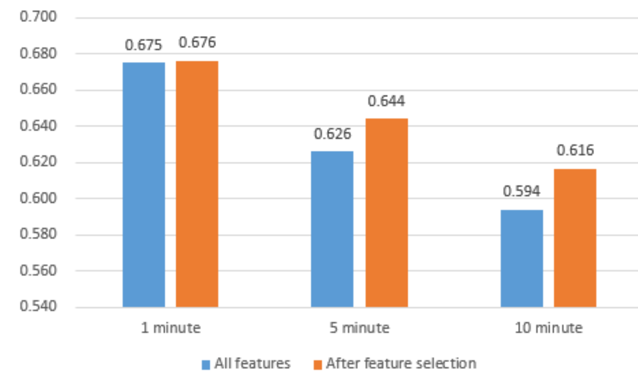

Figure 3: Performance of Random Forest after feature selection.

Performance in different time window. As shown in in Figure 3, the F1-score of the 1-minute time window outperformed the 5- and 10-minute counterparts. One explanation was that the longer the elapsed time of these events from the time of notification arrival, the weaker (or less accurate) the association between the notification and the specific sender the users could think of.

2.3 Features that selected in different time window

Recent Screen-On Event and Actions with the Phone. As shown in Table 3, *ExistScreenOn* and *ExistAction* outperformed other features in terms of importance scores computed by the Gini Importance[2], among all the time windows. In addition, in the remaining seven features (excluding *ExistScreenOn* and *ExistAction*), most features (six out of seven) were related to the recent app usage on

Table 3: Features we selected after feature selection. Score here is the score of feature importance.

Time window	Features	Score
1 minute	ExistScreenOnInMinute1	0.347
	ExistActionInMinute1	0.262
	ExistMessageNotiInMinute1	0.201
	UseSameMsgCateAppMinute1	0.19
5 minute	ExistScreenOnInMinute5	0.415
	ExistActionInMinute5	0.304
	UseSameMsgCateAppMinute5	0.195
	UseSameMsgPackageAppMinute5	0.0858
10 minute	ExistScreenOnInMinute10	0.348
	ExistActionInMinute10	0.318
	UseSameMsgPackageAppMinute10	0.151
	UseMsgAppMinute10	0.102
	UseSameMsgCateAppMinute10	0.081

[2] https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#giniimp

the phone. These results indicated that the existence of recent 1) screen-on events, 2) user-action events, and 3) app usage events, were predictive of whether or not users could accurately speculate the sender of the notification. These features are also similar in the sense that they both indicate the users' recent attention on the phone. This indicates that users were more likely to associate the arriving notification with a specific sender when they were recently attended to the phone.

***UseSameMsgCateApp* was also selected in all time windows**. For a similar reason, we think one explanation is that users were likely to associate the incoming notification with the recent activity they performed on the phone (e.g. sending a message, reading social media feeds). As a result, when the notification from the same kind of applications arrived, users were more likely to associate the notification with that activity, and perhaps also the people involved in that activity.

***ExistMessageNoti* was selected in 1 minute window**. We suggest *ExistMessageNotiInMinute1* is an important factor because users tended to speculate the same sender of the last notification they just read [1]. If users did speculate the sender of the current notification when the last notification was within one minute, then the sender of the last notification was very likely to be the one the user associated the current notification with. On the other hand, it is interesting to observe that, *ExistMessageNoti* was not selected in other time windows. This implies that the association triggered by the notification arrival decayed quite quickly. It is perhaps users may receive numerous notifications in larger time windows, and thus their association with the previous senders was likely to be interfered by the occurrence of new notifications.

Features related to recently used app. *UseSameMsgPackageApp* was added in both the 5-minute and the 10-minute time windows, which is absent in the 1-minute time window. We think this may be because users would need to recall the recently used app as another clue to help them speculate about the sender. As a contrast, for the 1-minute time window, users might not need this information for judging the sender if the notification arrives immediately after they had just used the phone. This might explain why *UseSameMsgPackageApp* did not contribute significantly for the 1-minute time window. However, since this is only our speculation about the possible explanation, further research would be needed.

3 DISCUSSION

Our results show that several events are crucial to whether users could judge the sender correctly or not, including 1) whether the users had physically acted on or attended to the phone; 2) whether the users received any relevant notifications; and 3) how far these events occurred prior to the notification. Interestingly, many of these factors are also found to be predictive of users' attentiveness [5]. Yet, this is not surprising, as [1] also showed that users' speculation of the notifications could affect their subsequent attendance to them. On the other hand, while [1] also mentioned that users' perceived temporal patterns of their notifications and the current context also offer important clues for users to form speculations, our features, which are limited to the information collectible on the phone, unfortunately, could not convey the value of these factors. We suggest future work considers these factors when building the models. There are also other features we did not explore, such as the identity of the contacts the user interacts with, such as whether the sender of the current notification is the latest person the user interacts with and the alerts that came with each notification. We suggest future work also explored these features.

4 CONCLUSION

This research aims to build a machine learning model that can predict the moments at which users cannot correctly judge the sender of the notifications prior to attending to them. The motivation comes from the prior work that if the speculation is successful and correct, users can selectively attend to the arriving notifications. Using the dataset collected from [1], which consisted of users' speculation behaviors of 667 notifications, we found Random Forest could achieve an F1-score (0.676), with recall: 84.39%, and precision: 56.78% in predicting these moments. Furthermore, we show the top contributing features for predicting these moments. Our results suggest that assisting users at these moments is promising, since predicting these moments are viable. On the other hand, the models we built did not achieve high precision, suggesting a high portion of false positive, i.e. users being able to judge the sender while being recognized by the model as not able to. More investigation and research efforts are needed to improve the prediction.

ACKNOWLEDGMENTS

This research was supported in part by the Ministry of Science and Technology, R.O.C (107-2218-E-009 -030 -MY3).

REFERENCES

[1] Yung-Ju Chang, Yi-Ju Chung, and Yi-Hao Shih. 2019. I Think It's Her: Investigating Smartphone Users' Speculation about Phone Notifications and Its Influence on Attendance. 1–13. https://doi.org/10.1145/3338286.3340125

[2] Joel E. Fischer, Nick Yee, Victoria Bellotti, Nathan Good, Steve Benford, and Chris Greenhalgh. 2010. Effects of Content and Time of Delivery on Receptivity to Mobile Interruptions. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services* (Lisbon, Portugal) *(MobileHCI '10)*. Association for Computing Machinery, New York, NY, USA, 103–112. https://doi.org/10.1145/1851600.1851620

[3] Hao-Ping Lee, Kuan-Yin Chen, Chih-Heng Lin, Chia-Yu Chen, Yu-Lin Chung, Chien-Ru Sun, and Yung-Ju Chang. 2019. Does Who Matter? Studying the Impact of Relationship Characteristics on Receptivity to Mobile IM Messages. https://doi.org/10.1145/3290605.3300756

[4] Abhinav Mehrotra, Veljko Pejovic, Jo Vermeulen, Robert Hendley, and Mirco Musolesi. 2016. My Phone and Me: Understanding People's Receptivity to Mobile Notifications. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 1021–1032. https://doi.org/10.1145/2858036.2858566

[5] Martin Pielot, Rodrigo de Oliveira, Haewoon Kwak, and Nuria Oliver. 2014. Didn't You See My Message? Predicting Attentiveness to Mobile Instant Messages. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 3319–3328. https://doi.org/10.1145/2556288.2556973

[6] Alireza Sahami Shirazi, Niels Henze, Tilman Dingler, Martin Pielot, Dominik Weber, and Albrecht Schmidt. 2014. Large-Scale Assessment of Mobile Notifications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 3055–3064. https://doi.org/10.1145/2556288.2557189