Adaptive Modelling of Attentiveness to Messaging: A Hybrid Approach

Pranut Jain
University of Pittsburgh
Pittsburgh, PA, USA
pranut@cs.pitt.edu

Rosta Farzan
University of Pittsburgh
Pittsburgh, PA, USA
rfarzan@pitt.edu

Adam J. Lee
University of Pittsburgh
Pittsburgh, PA, USA
adamlee@cs.pitt.edu

ABSTRACT

Identifying instances when a user will not able to attend to an incoming message and constructing an auto-response with relevant contextual information may help reduce social pressures to immediately respond that many users face. Mobile messaging behavior often varies from one person to another. As a result, compared to a generic model considering profiles of several users, a personalized model can capture a user's messaging behavior more accurately to predict their inattentive states. However, creating accurate personalized models requires a non-trivial amount of individual data, which is often not available for new users. In this work, we investigate a weighted hybrid approach to model users' attention to messaging. Through dynamic performance-based weighting, we combine the predictions of three types of models, a general model, a group model and a personalized model to create an approach which can work through the lack of initial data while adapting to the user's behavior. We present the details of our modeling approach and the evaluation of the model with over three weeks of data from 274 users. Our results highlight the value of hybrid weighted modeling to predict when a user cannot attend to their messages.

CCS CONCEPTS

• **Human-centered computing → User models**; *HCI theory, concepts and models.*

KEYWORDS

user clustering, inattentiveness, messaging, adaptive modelling

ACM Reference Format:
Pranut Jain, Rosta Farzan, and Adam J. Lee. 2019. Adaptive Modelling of Attentiveness to Messaging: A Hybrid Approach. In *27th Conference on User Modeling, Adaptation and Personalization (UMAP '19), June 9–12, 2019, Larnaca, Cyprus.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3320435.3320461

1 INTRODUCTION

Buzz! A message arrives on your mobile device in the middle of a meeting and with it comes a familiar sense of pressure and obligation to respond. Research shows that there is an expectation of fast responses associated with messaging applications [25, 34]. Recipients are keenly aware of these expectations which, if not met, can cause negative feelings in the sender and affect social relationships [10, 18]. Often, the recipient apologizes and explains delays in responding with reasons for their unavailability; e.g., "sorry - (got a) phone call" [45]. To help reduce the pressure on recipients and manage expectations on these types of platforms, an intelligent mobile messaging application that is aware of the user's context could proactively respond on behalf of the user, explaining their unavailability to the sender during inopportune instances.

The first step to *explaining* user inattentiveness is *accurate prediction* of it. Attentiveness to messaging is defined as the degree to which a user is paying attention to incoming messages [34]. A user is *inattentive* to messaging if he or she is unaware of an incoming message or its relevant details such as part of content and sender. Whether the user responds to the incoming messages or not can further depend on factors such as the message content and sender [28]. Our goal is to accurately identify instances when the user is not paying attention to incoming messages and thus we focus on modelling *attentiveness* to messaging rather than *responsiveness*.

One approach for modelling attentiveness is to aggregate data from a number of users to train a general model that can identify common messaging patterns with the assumption that these patterns are widely applicable to most users [34]. However, smartphone usage has been found to vary among people and may not be general [1, 46, 47]. That is, two users may act differently in similar contexts. Therefore, a more accurate approach would be to build personalized models to predict future instances of unavailability based upon users' own prior behavior. Previously, studies showed improved performance by adding personalization into the modelling approach for tasks like call-availability prediction [15, 32] and interruptibility prediction [30, 33, 40]. At the same time, to be able to build an accurate personalized model, sufficient usage data is required for each user. For new users, personalized modelling approach suffers the 'cold start' problem [42]. Due to the lack of sufficient initial data a personalized model can perform even worse than a general model [19]. Researchers have explored several approaches to address the cold-start problem. One such approach has been to consider group-based modeling; i.e., use only the limited amount of information about a target user, identify a cluster of users similar to the target user, and use the behavior of this group as the basis for modeling [26].

As previously explored in the field of user-modeling, group-based modeling approaches are useful for supporting users of adaptive systems when information about individuals is not available or collecting such information is not desirable; e.g., collecting privacy sensitive information [44]. In such approaches, users are often

UMAP '19, June 9–12, 2019, Larnaca, Cyprus

ACM ISBN 978-1-4503-6021-0/19/06…$15.00
https://doi.org/10.1145/3320435.3320461

clustered based on all available information including demographics and users' interaction behavior with the system. Consequently, the same recommendations are provided for all members of the group. However, group-based personalization models can face three challenges: (1) including information beyond the implicit users' interaction with the system such as demographic information can introduce additional barriers such as privacy concerns associated with collecting demographic information or requiring the users to explicitly provide additional information; (2) the performance of the model can depend highly on the accuracy of the clustering methods and set of features used in the clustering approach; and (3) using a group based model after enough personal information is available can lead to unnecessary sub-optimal performance.

In this work, we present our approach for building an adaptive hybrid weighted model that attempts to address these challenges in the context of predicting users' inattentiveness to mobile messages. We first present that in context such as mobile messaging where rich user-interaction data is available, a user clustering approach based on interaction and usage data can outperform clustering approaches based on users' characteristics such as age and gender. That is, in such context, there is no need to collect such additional information. We then describe our hybrid model of users' inattentiveness that is a weighted aggregate of general, group-based, and personalized models. We present our results of an evaluation analysis of this hybrid model as compared to each of the separate models. Our results highlight the importance of the ensemble model to achieve greater performance in predicting the inattentive state and solving the 'cold-start' problem.

Our work extends prior research in the field of user modeling by presenting a hybrid modeling approach for tasks that are highly context dependent and unstable over time. Our paper provides a detailed description of the modeling approach, supporting future researchers in replicating and extending our work. Furthermore, our modeling approach provides the first step towards building an intelligent messaging agent in which the basis of prediction allows to identify specific contextual characteristics explaining individual users' inattentiveness.

2 RELATED WORK

Our research is informed by three major areas of relevant work: (1) prior research that has explored the extent of social pressure associated with mobile messaging; (2) prior research in modelling users' behavior; and (3) prior research in user clustering. Below, we provide a summary for each area.

Social Pressure and Expectations in Mobile Messaging. It has been shown that senders expect the recipients to respond quickly to their messages, and at the same time recipients perceive the same pressure, at an even higher level [25]. A survey of mobile users to understand message senders' interpretation of no response to their messages shows that a large percentage of senders interpret it negatively. It was found that only 24% of the senders deem a recipient as 'is busy' while 15.4% respondents speculated that the recipient 'is pointedly ignoring me' or the recipient 'maybe in trouble' (5.7%), among other reasons [18]. These speculations are further fueled by availability indicators provided by messaging applications. Pielot et al. [34] showed that cues like 'last-seen' time are not only weak indicators of a user's attentiveness, but also create social pressure in the recipient and raise privacy concerns. A web-based survey of 945 users showed that 46.6% of respondents completely turned off the last-seen feature in WhatsApp owing to privacy concerns [37]. This body of research highlights the high degree of pressure associated with mobile messaging and a lack of effective solutions to support users of messaging applications.

Modelling Users' Availability. There has been a significant amount of work in modelling a user's availability in the context of mobile usage, including availability with respect to messaging [3, 13, 34], or phone-calls [32, 41], as well as modeling user behavior to predict opportune moments to send notifications to minimize interruptions [27, 30, 51]. Most closely to our work, Pielot et al. [34] showed that using contextual data from the user's device like ringer mode, screen status and proximity status can be used to model their attentiveness to messaging. Their generic model, aggregating data from 24 users collected for over 2 weeks, achieved 70% accuracy in predicting a user's attentiveness state. Avrahami et al. modelled a user's *responsiveness* to instant messages in a desktop environment. Using (1) IM based features like 'is message window open?' and 'buddy status'; and (2) Desktop features like 'Previous app in focus' they achieved an accuracy as high as 90% by utilizing decision trees to predict whether a user will respond to an incoming message within 5 minutes. Our work extends the prior research in this area by investigating a group-based and personalized modelling of availability and further evaluates how these models can be integrated for an optimal hybrid model.

Detecting User Groups. Detecting groups of similar users has been studied extensively in the context of recommender systems for collaborative filtering and particularly to tackle the 'cold start' problem for new users who have not yet rated enough items to get accurate recommendations [17, 38]. Pearson Correlation has been used as one of most common approaches in calculating users' similarity and clustering users [24, 36]. In one approach, the cluster membership was identified by suggesting items to the user to rate which provide the maximum information gain to distinguish branches of a decision tree whose leaf nodes represent user clusters [36]. Other approaches have involved clustering users into groups based on their like or dislike of a specific set of items, such as movies. Within these groups, rating for unrated items were predicted and the least error item was selected for recommendation [16]. There have been other approaches to cluster users based on information available about them in addition to their behavior in the system, such as clustering users by their demographic characteristics [31]. User clustering has also been applied to other tasks such as churn prediction. Yang et al. grouped users based on their daily activity and ego network on SnapChat into interpretable clusters to leverage correlation between users to predict social media churn for a new user with limited behavioral data [50]. In this work, we explore demographic vs. usage behavior clustering methods and present how each of the methods can benefit our overall modelling goal.

3 METHODS

Here, we describe the dataset we used in this study, the types of features we extracted, our target variable and evaluation metrics.

3.1 Dataset Used

The dataset used in this work was extracted from smartphone sensor logs of 342 participants collected over an average period of four weeks. This data was collected as part of a study by Pielot et al. to predict opportune moments when a user would be willing to engage with the contents of a mobile notification [33]. The sensor events in the log correspond to (1) change-based events such as change in screen status from *on* to *off* or *unlocked*; (2) usage-based events such as number of incoming messages, notifications and phone calls; and (3) state-based events captured every 10 minutes such as battery state and connectivity (e.g., cellular, WiFi) state.

From these sensor logs for each user, we extracted incoming messaging notification events along with the device and user state at the time of the notification. Each row of the extracted dataset for each user corresponds to an incoming messaging notification. We focus only on notifications generated by WhatsApp messenger [1] since they make up 92% of all notifications by communication category applications in this dataset.

Our final dataset comprised of 1,375,359 notification instances from 274 participants spanning an average time-period of 3 weeks.

3.2 Feature Set

Our feature set includes information about the user and device context at the time of an incoming message. We extracted a total of 72 features belonging to four categories corresponding to

- **Current state of the device**, for example, device orientation (portrait/landscape) and semantic location of the user (home, work or passing), current activity (on foot, cycling)
- **Time elapsed since last event**, for instance, time since an application was last opened or an outgoing call was made
- **Device usage in the last hour**, for instance, number of notifications received and network data transmitted.
- **Device usage in the current day**, for instance, percentage of time spent at home or at work and total battery time.

3.3 Target Variable

The target variable is the user's attentiveness state at the time of an incoming message. A user is *attentive* to messaging if he or she attends to an incoming message within a threshold of time. A notification can be attended to by (1) Opening the application which generated it; (2) Accessing the notification center which provides relevant details about a notification like sender and part of the content; (3) Accessing the notification on another device [13]. We picked the threshold for attentiveness based on the median time to attend a messaging notification, which in our dataset, averaged across all users was 5.10 minutes.

We used accuracy and F-measure for the inattentive class to evaluate model performances.

4 MODELLING ATTENTIVENESS

In this section, we describe our approaches for producing both general and personalized models of user attentiveness to messaging. In the following section, we explore approaches for building group-based models that interpolate between these two extremes.

[1]WhatsApp, https://www.whatsapp.com/

4.1 General Model

We constructed our general model based on the aggregated data from all the users to form a single model [34]. We utilized a scalable gradient boosting decision tree approach using XGBoost [8] to build the model. We set parameters 'max_depth' which is the maximum depth of the tree to '5' and 'min_child_weight' which specifies the minimum weight to further partition the tree to '20' based on the results from the tuning process. Other parameters had an insignificant impact on the model performance during evaluation.

To evaluate the general model, we employed 10-fold grouped cross-validation approach to ensure that the instances for a user are not split across training and testing folds. This evaluation provides an estimate as to how the model would perform for new users for whom training data is not yet available. The general model achieved an accuracy of 72.28% and F-measure for the inattentive state of 0.651. Table 1 shows the top features identified by the general model ordered by the 'gain' provided to the model.

4.2 Personalized Models

We built individual personalized models by utilizing only a user's own messaging data. We trained individual models using XGBoost with default parameters and number of boosting iterations set to '20'. Notification data is time-ordered and might contain sessions of fast message exchange (instant messaging sessions [3]). This creates a dependency structure between instances which is not accounted for in most machine learning algorithms. Thus, randomized cross-validation would tend to overestimate the model performance while sequential validation would underestimate it [12, 39]. To be able to correctly evaluate individual models, we instantiated blocks of notification instances which arrived close to each other (for example, within 15 seconds). 10-fold grouped cross-validation was then performed making sure the instances in a given block were not used in both training and testing folds.

On average, personalized models achieved an accuracy of 84.21% and F-measure for the inattentive state of 0.744. Table 1 lists what fraction of personalized models had the same top feature from the general model in their top-5 features. It was observed that only 40% of the personal models had the same top feature of the general model in their top-5 features, suggesting that personalized models give more weight to variables relevant to the specific user being modelled.

5 USER CLUSTERING APPROACHES

In this section, we present our approaches to user clustering based on two common set of features: (1) users' demographics and (2) users' daily interaction and usage behavior. For each cluster of users that was identified, we built a group attentiveness model using aggregate data of the members of the cluster utilizing similar approach as the general model. To evaluate each grouped model, we performed grouped cross-validation as discussed in section 4.1 to assess the group model performance on its associate members.

5.1 Demographics based clustering

Prior research suggests that age and gender have a significant effect on smartphone usage patterns [1]. Messaging behavior in particular, shows high variance across different age and gender groups [29].

Table 1: Top features identified by the general model along with their score (gain) along with what fraction of users have that feature in their top 5 features of their personalized individual model and which group models have it in their top-10 features.

Feature name	Description	General Model feature-score	Personalized Models: fraction of users	Group Models
timeSinceLastOpenApp	# ms since any app opened	7578	40.31%	1,2,3
Screen_Value	current screen status	2448	41.47%	1,2,3
timeSinceWhatsAppOpened	# ms since any whatsapp opened	1178	17.44%	1,2,3
timeSinceScreenChanged	# ms since screen changed	843	22.48%	1,2,3
Charging_Value	whether the device is charging	540	2.32%	1,2,3
HourOfDay	current hour of the day	466	1.55%	2
App_Value	current foreground app	427	4.26%	1,2
CellTower_GSMErr	amount of signal error	402	0%	-
perc_noloc	% time device unable to get loc	394	10.65%	-
timeSinceNotifCenter	# ms since notif center accessed	391	10.46%	2

Thus, demographics-based grouping forms a good starting point to identify similar groups of users.

5.1.1 Age-based clusters. Users' ages in our dataset range from 18 to 66 years. To find appropriate groups of users based on their age, we used Jenks natural breaks optimization to find breaks in the age distribution of the dataset. The number of breaks set to five resulted in a goodness of variance fit (GVF) value of 0.92. The resulting five categories of age groups along with their attentiveness model performance are summarized in Table 2.

Only the attentiveness model for age group 44–50 showed significantly better performance in detecting the inattentive state than the general model which can be attributed to the fact that members of this group were less attentive to messaging compared to other groups (52% inattentive vs 48% attentive instances).

5.1.2 Gender-based clusters. Gender-based groups along with their attentiveness model performance are summarized in Table 2. Gender 0 makes up 47% of the dataset and its attentiveness model showed minor improvement over the general model, while Gender 1 model showed lower performance compared to the Group 0 model and the general model.[2]

Overall, our results show that when considering inattentiveness to messaging notifications, group-modeling based on age and gender does not provide an improvement over our general model.

5.2 Usage-based clustering

Our second clustering approach focuses on clustering users based upon their daily smartphone usage profile. To do so, we first explored what set of usage features can most effectively discriminate users into coherent clusters. Research has shown that variation in user behavior can observed in different dimensions including location semantics [23], application usage [49, 52], movement patterns and connectivity [46]. Thus, we did not make any prior assumptions as to which behavioral categories will exhibit the maximum variance among users. We extracted an exhaustive feature set from all types of sensor events comprising of multiple categories such as (1) context-based features like time spent at home, at work, and commuting; (2) device-based features like the number of times device was plugged in, screen state changed events, and device orientation changed events; and (3) communication-based features like the number of phone calls received, duration of incoming calls, and number of messages received.

The final behavioral matrix X_i is of the shape $N \times K$ where N (=274) is the number of users and K (=52) is the number of feature dimensions. Each row of matrix X_i summarizes a user's daily behavior on average. We did not include any demographics-based features in the feature set.

5.2.1 Clustering Approach. We used a Bayesian Gaussian Mixture Model (BGMM) utilizing variational inference [2, 4] to estimate the membership of data points to a cluster. BGMM can be used as an unsupervised clustering approach which has the advantage of not needing a pre-defined number of clusters as it chooses the optimal number of components to best fit the data. We set each component to have its own general covariance matrix allowing them to adopt to any shape and position. The number of expectation maximization iterations was set to 200 with number of initializations set to 10. Upon fitting the model to the behavioral matrix X_i three components or user clusters were returned.

5.2.2 Cluster Analysis and Interpretation. As presented in Table 2, Cluster 1 included 137 users, Cluster 2 included 87 users, and Cluster 3 included 50 users. To visualize the identified clusters along dimensions of high variability and find correlated features we conducted principal component analysis [20] of the behavioral matrix X_i. We standardized each feature f_i of X_i before computing the principal components. The top 3 principal components along with associated features are summarized in Table 3.

Principal component 1 accounts for 15% of the variance of the data. The three main features comprised in PC-1 are number of communication notifications dismissed, number of applications opened, and number of times notification center was accessed. The second principal component makes up 9% of variability in the data and is related to the features such as number of incoming calls, time spent on incoming calls, and number of missed calls. The third principal component captures 6% variability of the data and comprises of features such as time connected to mobile data, amount of time not connected to a WiFi network, and number of

[2]Our data represents gender only as 0 and 1 without association to any specific gender

Table 2: Grouping Summary. The accuracy and F-measure (inattentive) are computed by evaluating the model formed from the aggregate data of the members of the group.

	Group	Users	Accuracy	F-measure
Age	18-26	50	75.40	**0.660**
	27-35	78	69.84	0.574
	36-43	57	69.00	0.588
	44-50	50	70.46	**0.697**
	51-66	39	63.45	0.611
Gender	0	128	72.25	**0.664**
	1	146	72.47	0.634
Daily Behavioral	Cluster 1	137	72.60	**0.679**
	Cluster 2	87	70.59	0.589
	Cluster 3	50	71.14	**0.704**

Table 3: Top three Principal Components for the daily usage behavioral matrix X_i

Principal Component	Feature	Score
PC-1 (variance_ratio = 0.155)	num_comm_dismissed	+0.286
	num_app	+0.284
	num_notifcenter	+0.277
PC-2 (variance_ratio = 0.091)	num_incomingcall	+0.351
	time_incall	+0.337
	num_missedcall	+0.291
PC-3 (variance_ratio = 0.063)	time_data_conn	-0.348
	time_wifi_noconn	-0.307
	num_outgoingcall	-0.302

outgoing calls. Principal component 1 based on comprised feature weights, signifies variability between users in how actively they check and interact with their phone. Principal component 2 is linked to how actively a user is engaged in phone calls and principal component 3 signifies variability between users based on their network connection status.

Cluster assignments with respect to the top three principal components is shown in Figure 1. A clear distinction between the three identified clusters can be observed on both PC-1 vs PC-2 (Figure 1a) and PC-3 vs PC-1 (Figure 1b) plots. On further analysis of the three clusters, it was observed that cluster 2 users show comparatively **more active** use of their device as they frequently check their phones and open higher number of applications throughout the day. Whereas, cluster 1 users are **less active** users who receive lower number of notifications per day and in general have lower interaction with their device. Cluster 3 users are moderately active with regards to interaction with their phone but are **active callers** as they receive and make relatively more number of phone calls. They also show more time on a data connection rather than on WiFi connection. Further, they spend more time travelling as they have higher daily on-foot, cycling and in-vehicle times which explains the longer periods on data connection.

5.2.3 Group-based modelling based on usage clusters. We modelled attentiveness for each group of users identified by the clusters. As presented in Table 2, clusters 1 and 3 showed a significant improvement on the mean F-measure for the inattentive state while Cluster 2 exhibited a much lower average F-measure compared to the general model. Since cluster 2 users are more active device users, it becomes harder to detect their inattentive states which is evident by the imbalance in their class distribution (39% inattentive vs 61% attentive states).

Previously it has been noted that recency in receiving and making phone calls has been directly linked to a user's availability [33, 35] which would explain the easier detection of the inattentive state for users in cluster 3 which showed a significant improvement in F-measure (inattentive) over the general model. Table 1 also shows which group models share the same top features as the general model in their top 10 features.

6 ADAPTIVE WEIGHTED MODELING

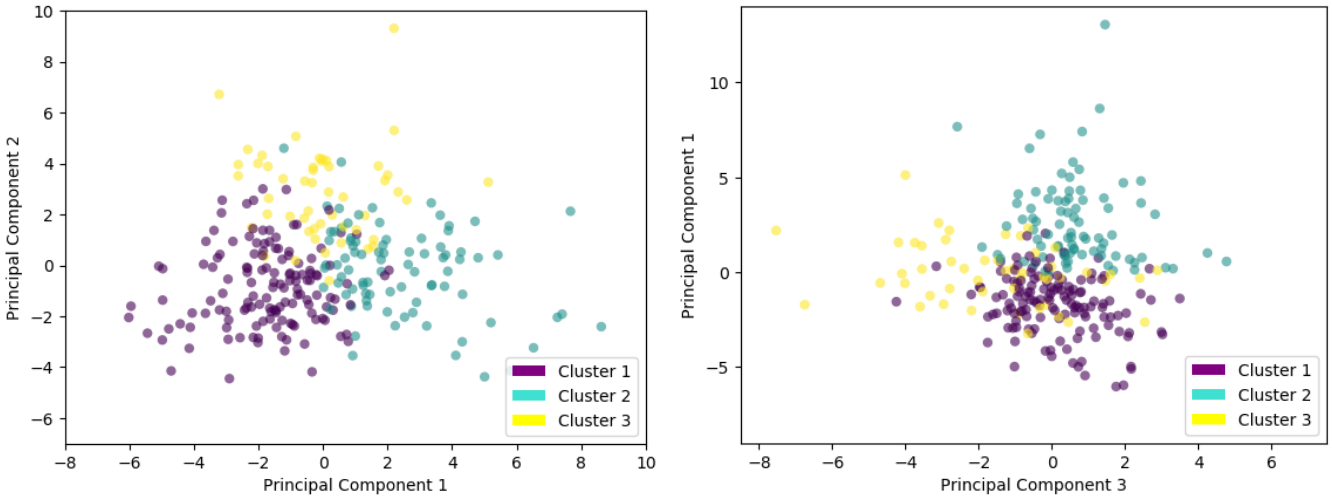
As discussed before, while personalized models can provide more accurate modeling of an individuals' behavior as basis for prediction, they can only achieve that when enough user data is available. In face of insufficient personal data, a general model can outperform a personalized model and the group-based model, given correct association for a new user to a behavioral cluster, would perform event better. Our results of usage-based clustering analysis show that the group-based attentiveness model for two out of the three user groups outperforms a general model for predicting a user's inattentive state when compared to a general model. Therefore, if a new user demonstrates similar daily behavior as users in these two groups, predicting their inattentive state should be done based on their matching group model rather than a general model considering all the users. However, relying on a single type of model may not be the best approach as: (1) depending upon usage behavior and lack of initial data, a general model may perform the best for some users; (2) behavior based clustering approach requires at least a day of usage data to detect the behavioral group for a user which may not be representative of the user's behavior as group membership could change as more data becomes available; (3) even with adequate amount of data, a personalized model would require time to adapt to sudden changes in user's behavior and environment.

As a result, an accurate modeling of user attentiveness requires different modeling approaches at different stages. Rather than treating this issue as a model selection problem, we approach it by building a hybrid model that aims to integrate predictions from multiple models to be able to adapt to the situations mentioned above without relying solely on the amount of data available. Algorithm 1 goes over how the adaptive model works. Our hybrid modeling approach is discussed in detail below.

Given a data point x_i, its class y_i can be determined by

$$y_i = \sum_{c \in C} w_c * y_c(x_i) \quad (1)$$

where $C = \{cluster, general, personalized\}$ is the set of models in use, w_c is the weight associated with model c and ranges between $\{0, 1\}$ and $y_c(x_i)$ is the class predicted by model c for the data point x_i. For modelling approaches that return the probability of each class for a given data point rather than the class value, we can



(a) PC-1 vs PC-2 **(b) PC-3 vs PC-1**

Figure 1: Plot comparing cluster assignments against Top 3 principal components

rewrite equation 1 with the probability value returned by the model for the inattentive class, $P_c(y_i = 0)$:

$$P(y_i = 0) = \sum_{c \in C} w_c * P_c(y_i = 0) \tag{2}$$

We can then consider that if $P(y_i = 0) > 0.5$, set the class as *inattentive* or adjust that threshold to different values for more relaxed or more conservative models.

To set the weights w_c assigned to each model, the basic approach can be to set them to a pre-computed static value or as a function of the amount of data available for a user since heuristically as more data becomes available the weights for the personalized model should be increased while reducing the weights for the group and general models. However, statically set weights would not take into consideration sudden changes in user behavior which can affect the model performance.

Therefore, to address this limitation, we propose a dynamic approach to update the model weights based on how well a model performed recently for a given user. Previously, accuracy of prediction through RMSE (Root Mean Squared Error) has been used to derive weights for classifiers in the ensemble model [5, 48]. This method of accuracy-weighted voting does not work well for unbalanced datasets [7]. Hence, instead, we use F-measure (for inattentive class) to determine the 'fitness' of a model in the ensemble [7].

Let w_c^{t+1} be the weight of model to be used at the next time step $t + 1$, then

$$w_c^{t+1} = \frac{f_c^t + \alpha(\Delta f_c^t)^3}{\sum_{m \in C} f_m^t + \alpha(\Delta f_m^t)^3} \tag{3}$$

where f_c^t is the performance of model c in terms of f-measure for the inattentive state at the current time-step t, α is a constant and Δf_c^t is the change in the performance of model c from previous time-step i.e.

$$\Delta f_c^t = f_c^t - f_c^{t-1} \tag{4}$$

The denominator normalizes the weight to be in range $\{0, 1\}$. We take the cube of Δf_c^t to emphasize larger gains while keeping the *sign* of the change in performance. As observed from equation 3, the weight of the model for the next time-step only depends upon the model's performance in the current time-step and the change in performance from the previous time-step. The term $\alpha(\Delta f_c^t)^3$ will either reward or penalize the model based on the change in its performance. The parameter α can be tuned based upon the granularity of the time-step t. If the weights are updated per instance basis, then α should have a lower value while with day-to-day weight update, it should be set to a higher value.

This type of weight assignment scheme not only allows the adaptive model to adjust to amount of user data available but also to adopt to users' most recent behavior. For instance, a user's messaging patterns might change while on vacation. The personalized model might not have observed the user's behavior in this new environment in the past and thus its performance would likely suffer. Detecting this drop of performance, the adaptive model would penalize its weight for the next time-step until the personalized model adapts to this new environment.

Identifying most important features in a model is often essential to improve the model or in our case, form explanations for users' inattentive state. To compute the relative importance of features, we multiply individual feature scores of each model with the model weight and then pick the top k scoring features. The feature scores can be the 'gain' provided to the model by the feature or other metrics such as information gain ratio.

7 EVALUATION

The evaluation process that we used has been summarized in Algorithm 2. The objective of the evaluation was to simulate multiple modelling approaches for a new user and get an estimate of how each of them perform as more data becomes available over time.

Define:
clu, gen, per = group, general, personal models
f^* = set of f-measures of each model
day_usage = aggregate user behavior for the current day
day_instances = message instances current day
Input : x: a new instance of incoming message
Output: $state$: attentiveness state
begin
/* check if a new day has begun */
if $getday() \neq current\_day$ **then**
f^* = compute_models_performance(y_preds, y_true)
clu = get_cluster(day_usage)
w^* = update_weights(f^*) using eq. 3
per = update_personalized_model(day_instances)
reset f^*, day_usage and day_instances
current_day = getday()
$P_{gen}(y_i = 0) = gen(x_i)$
$P_{clu}(y_i = 0) = clu(x_i)$
$P_{per}(y_i = 0) = per(x_i)$
$P(y_i = 0)$ = combine predictions using equation 2
if $P(y_i = 0) > 0.5$ **then**
state = $inattentive$
else
state = $attentive$
y_preds.add(model, state)
return state
end

Algorithm 1: Adaptive Modelling

foreach *user* $u \in U$ **do**
/* train general model without user u */
$gen_u = train_{gen}(data - data_u)$
/* perform clustering without user u */
clusters = user_clustering(U-u)
for $d \in range(1, k)$ **do**
/* get cluster membership based on average cumulative daily data for day d */
user_cluster = get_cluster(clusters, $daily_u^d$)
/* train group model with similar users data */
$clu_u = train_{clu}(user\_cluster)$
train_size = d/k
/* get user data split by day, 10 folds */
train_data, test_data = groupCV(train_size)
$per_u = train_{per}(train\_data)$
if $d = 1$ **then**
/* Initialize model weights using training data for day 1 */
w^* = initialize_weights(train_data)
$pred_{gen} = gen_u(test\_data)$
$pred_{clu} = clu_u(test\_data)$
$pred_{per} = per_u(test\_data)$
$pred_{adapt}$ = combine predictions using equation 2
f^* = compute_models_performance(y_preds, y_true)
w^* = update_weights(f^*) using eq. 3
end
end

Algorithm 2: Evaluation process

For each user, the amount of data available was gradually increased in one-day increments. The available data for the user was split in proportion of d/k where d is the number of days of data to use for training and k is the total number of days of data available for that user. This forms the training set for the personalized model and the remaining $(1 - \frac{d}{k})$ data becomes the testing set. For consistency in the number of users during the evaluation process, we only considered users with at least 18 days of messaging data available in our dataset which made up 79% (216) of all users. The general model was trained as discussed in Section 4.1 while not including the data of the target user. Similarly, cluster detection as discussed in Section 5.2 was performed to find and model user groups without including the target user. To determine initial cluster membership, only one day of usage data of the target user was utilized and as more data became available, cluster membership was re-evaluated. The general and group models were also evaluated on the same test data as the personalized model. The predictions of all three models were then combined as discussed in Section 6 to get the predictions for the hybrid weighted model. We repeated this process for each user in the dataset and averaged the performance of each model over all users for each day. The plot comparing the average model performance with increasing amount of available data in terms of number of days is shown in Figure 2. It can be observed that the personalized model performance is considerably low during the first few days due to the lack of training data. The general and group models show consistent average performance throughout the testing period. Group models, on average, slightly under-performed compared to the general model as the general model performed significantly better for one of the discovered clusters in detecting inattentiveness, bringing the average down.

The adaptive model performs better than all other models during the starting few days and eventually settles at personalized model performance. To assess what impact the group model has on the adaptive model, we included a plot of adaptive model performance without including the predictions of the group model. It can be observed that there is a performance drop until day 6, confirming that the group models provide a significant gain to the adaptive model for the initial few days. While a few days might not seem significant, but it should be considered that most users decide to utilize a new application based upon their initial experiences. A disappointed new user would likely not return to the application [21].

Figure 3, shows how the dynamically assigned model weights change over time as more data becomes available. This plot can also be interpreted as the relative model importance with respect to time. The weight for the personalized model increases sharply as more data becomes available and after day four, has more weight than the group and general models. The change in weights subsides

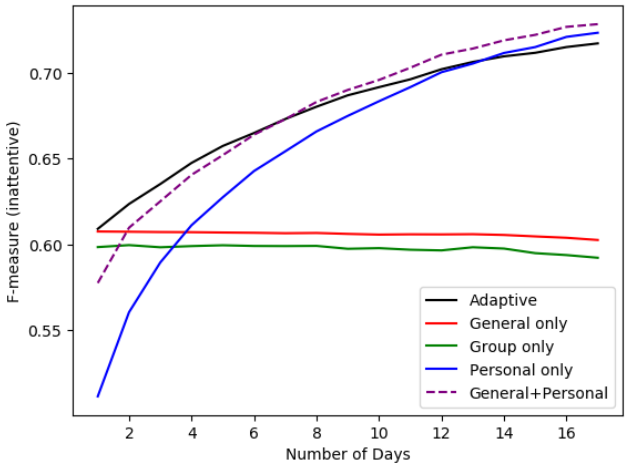


Figure 2: Comparing model performances based on days of data available

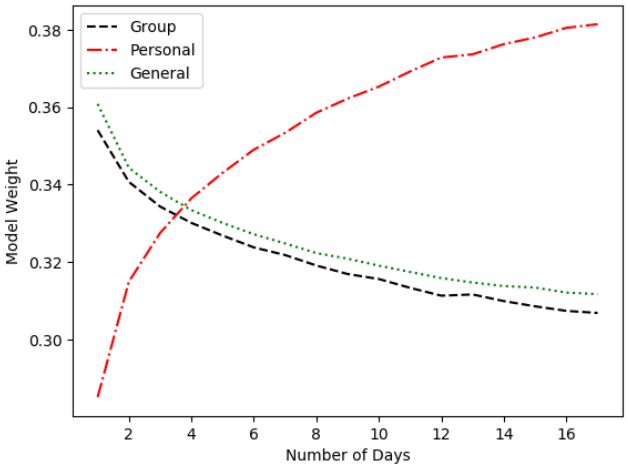


Figure 3: Change in model weights over time

around the 16-day mark with general at 0.312, cluster at 0.307 and personalized model at 0.381.

8 DISCUSSION AND FUTURE WORK

In this paper, we presented an approach for building an ensemble model to accurately predict instances when users are inattentive to messaging. We present how this hybrid approach can overcome challenges faced by different modeling approaches alone. Our approach allows for the model to adapt to user behavior as more data is collected by (1) considering a dynamic, usage-based clustering approach and (2) creating a hybrid weighted model that optimally combines information about the user being profiled with models of more general user classes.

Computationally, our approach involves three modelling stages. First, we must train the general model, which needs to be done infrequently unless the user population changes significantly. Second, we must maintain up-to-date group models, which requires identifying group memberships for individual users, as well as training group models. While the group membership for a user can change over time, the group model does not need to be retrained frequently. Third, we must update the personalized models regularly to be able to adapt to changes in a users' behavior and environment. In this work, we used a batch training approach, which required retraining the model again as more data became available. This frequent retraining not only takes up computation resources but also requires storing batches of user data which can subject the users to privacy compromises of their personal data. Another approach would be to use an online or incremental classifier [14, 30, 51]. Incremental approaches update the model per instance or in mini-batches and often do not require previously used training data while also reducing the training time significantly [6]. Though in a lot of cases, they do not perform as well as batch trained models [6, 9, 43].

Being able to accurately detect instances of inattentiveness is the first step towards the design of an intelligent messaging assistant to support users during moments of unavailability. Our next steps include generating textual auto-responses to explain a recipient's unavailability to the message sender. These responses can either be sent on a user's behalf automatically or provided to the user as a suggestion. Constructing such responses requires understanding what contextual factors are affecting a user's availability at the time of an incoming message. This information can be extracted from the user's attentiveness model which captures the user's messaging behavior. However, there are a number of challenges involved in this endeavor, particularly to ensure identification of accurate and *effective* [11, 22] information regarding the user's state and to ensure protection of user's privacy. Our future work aims to design, implement and evaluate an effective messaging support agent which is also sensitive to users' privacy concerns.

9 CONCLUSION

In this work, we took the first step towards the design of an intelligent messaging agent which can detect instances of unavailability for a user and act on their behalf reducing the pressure on them to keep checking their phone for unanswered messages. We evaluated multiple attentiveness modelling approaches and observed that personalized models suffer from the 'cold-start' problem, where performance is poor during initial periods of sparse user data. On the other hand, general models are not able to achieve high performance due to the diversity in usage and messaging behaviors amongst individuals. To tackle this issue, we grouped similar users together based on their daily usage behaviors to identify three groups, two of which performed better than the general model to detect inattentiveness for their members achieving F-measures of 0.679 and 0.704 respectively. Finally, we combined the predictions of all three types of models to create an adaptive hybrid model which outperformed all other modelling approaches during the initial days of evaluation and is flexible to changes in user behavior.

ACKNOWLEDGMENTS

The authors wish to thank Martin Pielot for providing the dataset used in this work. This work was supported in part by the National Science Foundation under awards CNS–1253204 and CNS–1814866.

REFERENCES

[1] Ionut Andone, Konrad Błaszkiewicz, Mark Eibes, Boris Trendafilov, Christian Montag, and Alexander Markowetz. 2016. How age and gender affect smartphone usage. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. ACM, 9–12.

[2] Hagai Attias. 2000. A variational baysian framework for graphical models. In *Advances in neural information processing systems*. 209–215.

[3] Daniel Avrahami and Scott E Hudson. 2006. Responsiveness in instant messaging: predictive models supporting inter-personal communication. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 731–740.

[4] David M Blei, Michael I Jordan, et al. 2006. Variational inference for Dirichlet process mixtures. *Bayesian analysis* 1, 1 (2006), 121–143.

[5] Dariusz Brzeziński and Jerzy Stefanowski. 2011. Accuracy updated ensemble for data streams with concept drift. In *International conference on hybrid artificial intelligence systems*. Springer, 155–163.

[6] Nikolay Burlutskiy, Miltos Petridis, Andrew Fish, Alexey Chernov, and Nour Ali. 2016. An Investigation on Online Versus Batch Learning in Predicting User Behaviour. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence*. Springer, 135–149.

[7] Nitesh V Chawla and Jared Sylvester. 2007. Exploiting diversity in ensembles: Improving the performance on unbalanced datasets. In *International Workshop on Multiple Classifier Systems*. Springer, 397–406.

[8] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 785–794.

[9] Sungjoon Choi, Eunwoo Kim, and Songhwai Oh. 2013. Human behavior prediction for smart homes using deep learning.. In *RO-MAN*, Vol. 2013. 173.

[10] Karen Church and Rodrigo De Oliveira. 2013. What's up with whatsapp?: comparing mobile instant messaging behaviors with traditional SMS. In *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services*. ACM, 352–361.

[11] Edward S De Guzman, Moushumi Sharmin, and Brian P Bailey. 2007. Should I call now? Understanding what context is considered when deciding whether to initiate remote communication via mobile devices. In *Proceedings of Graphics interface 2007*. ACM, 143–150.

[12] Thomas G Dietterich. 2002. Machine learning for sequential data: A review. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. Springer, 15–30.

[13] Tilman Dingler and Martin Pielot. 2015. I'll be there for you: Quantifying Attentiveness towards Mobile Messaging. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, 1–5.

[14] Ryan Elwell and Robi Polikar. 2011. Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks* 22, 10 (2011), 1517–1531.

[15] Robert Fisher and Reid Simmons. 2011. Smartphone interruptibility using density-weighted uncertainty sampling with reinforcement learning. In *Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on*, Vol. 1. IEEE, 436–441.

[16] Nadav Golbandi, Yehuda Koren, and Ronny Lempel. 2011. Adaptive bootstrapping of recommender systems using decision trees. In *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 595–604.

[17] Songjie Gong. 2010. A collaborative filtering recommendation algorithm based on user clustering and item clustering. *JSW* 5, 7 (2010), 745–752.

[18] Roberto Hoyle, Srijita Das, Apu Kapadia, Adam J Lee, and Kami Vaniea. 2017. Was my message read?: Privacy and Signaling on Facebook Messenger. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 3838–3842.

[19] Scott Hudson, James Fogarty, Christopher Atkeson, Daniel Avrahami, Jodi Forlizzi, Sara Kiesler, Johnny Lee, and Jie Yang. 2003. Predicting human interruptibility with sensors: a Wizard of Oz feasibility study. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 257–264.

[20] Ian Jolliffe. 2011. Principal component analysis. In *International encyclopedia of statistical science*. Springer, 1094–1096.

[21] Sang Chon Kim, Doyle Yoon, and Eun Kyoung Han. 2016. Antecedents of mobile app usage among smartphone users. *Journal of marketing communications* 22, 6 (2016), 653–670.

[22] Johannes Knittel, Alireza Sahami Shirazi, Niels Henze, and Albrecht Schmidt. 2013. Utilizing contextual information for mobile communication. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. ACM, 1371–1376.

[23] Min-Joong Lee and Chin-Wan Chung. 2011. A user similarity calculation based on the location for social network services. In *International Conference on Database Systems for Advanced Applications*. Springer, 38–52.

[24] Qing Li and Byeong Man Kim. 2003. Clustering approach for hybrid recommender system. In *null*. IEEE, 33.

[25] Lisa M Mai, Rainer Freudenthaler, Frank M Schneider, and Peter Vorderer. 2015. âĂIJI know youâĂŹve seen it!âĂİ Individual and social factors for usersâĂŹ chatting behavior on Facebook. *Computers in Human Behavior* 49 (2015), 296–302.

[26] Judith Masthoff. 2011. Group recommender systems: Combining individual models. In *Recommender systems handbook*. Springer, 677–702.

[27] Abhinav Mehrotra, Mirco Musolesi, Robert Hendley, and Veljko Pejovic. 2015. Designing content-driven intelligent notification mechanisms for mobile applications. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 813–824.

[28] 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*. ACM, 1021–1032.

[29] Christian Montag, Konrad Błaszkiewicz, Rayna Sariyska, Bernd Lachmann, Ionut Andone, Boris Trendafilov, Mark Eibes, and Alexander Markowetz. 2015. Smartphone usage in the 21st century: who is active on WhatsApp? *BMC research notes* 8, 1 (2015), 331.

[30] Veljko Pejovic and Mirco Musolesi. 2014. InterruptMe: designing intelligent prompting mechanisms for pervasive applications. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 897–908.

[31] Andre Luiz Vizine Pereira and Eduardo Raul Hruschka. 2015. Simultaneous co-clustering and learning to address the cold start problem in recommender systems. *Knowledge-Based Systems* 82 (2015), 11–19.

[32] Martin Pielot. 2014. Large-scale evaluation of call-availability prediction. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 933–937.

[33] Martin Pielot, Bruno Cardoso, Kleomenis Katevas, Joan Serrà, Aleksandar Matic, and Nuria Oliver. 2017. Beyond interruptibility: Predicting opportune moments to engage mobile phone users. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 91.

[34] 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*. ACM, 3319–3328.

[35] Martin Pielot, Tilman Dingler, Jose San Pedro, and Nuria Oliver. 2015. When attention is not scarce-detecting boredom from mobile phone usage. In *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*. ACM, 825–836.

[36] Al Mamunur Rashid, George Karypis, and John Riedl. 2008. Learning preferences of new users in recommender systems: an information theoretic approach. *Acm Sigkdd Explorations Newsletter* 10, 2 (2008), 90–100.

[37] Yasmeen Rashidi, Kami Vaniea, and L Jean Camp. 2016. Understanding SaudisâĂŹ privacy concerns when using WhatsApp. In *Proceedings of the Workshop on Usable Security (USECâĂŹ16)*.

[38] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. Recommender systems: introduction and challenges. In *Recommender systems handbook*. Springer, 1–34.

[39] David R Roberts, Volker Bahn, Simone Ciuti, Mark S Boyce, Jane Elith, Gurutzeta Guillera-Arroita, Severin Hauenstein, José J Lahoz-Monfort, Boris Schröder, Wilfried Thuiller, et al. 2017. Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography* 40, 8 (2017), 913–929.

[40] Stephanie Rosenthal, Anind K Dey, and Manuela Veloso. 2011. Using decision-theoretic experience sampling to build personalized mobile phone interruption models. In *International Conference on Pervasive Computing*. Springer, 170–187.

[41] Iqbal H Sarker. 2018. Silentphone: Inferring user unavailability based opportune moments to minimize call interruptions. *arXiv preprint arXiv:1810.10958* (2018).

[42] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*. Vol. 39. Cambridge University Press.

[43] Jeremiah Smith, Anna Lavygina, Jiefei Ma, Alessandra Russo, and Naranker Dulay. 2014. Learning to recognise disruptive smartphone notifications. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*. ACM, 121–124.

[44] Barry Smyth, Jill Freyne, Maurice Coyle, Peter Briggs, and Evelyn Balfe. 2003. I-SPYâĂŤAnonymous, community-based personalization by collaborative meta-search. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence*. Springer, 367–380.

[45] Amy Voida, Wendy C Newstetter, and Elizabeth D Mynatt. 2002. When conventions collide: the tensions of instant messaging attributed. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 187–194.

[46] Daniel T Wagner, Andrew Rice, and Alastair R Beresford. 2013. Device analyzer: Understanding smartphone usage. In *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*. Springer, 195–208.

[47] Tanja Walsh, Piia Nurkka, and Rod Walsh. 2010. Cultural differences in smartphone user experience evaluation. In *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia*. ACM, 24.

[48] Haixun Wang, Wei Fan, Philip S Yu, and Jiawei Han. 2003. Mining concept-drifting data streams using ensemble classifiers. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. AcM, 226–235.

[49] Qiang Xu, Jeffrey Erman, Alexandre Gerber, Zhuoqing Mao, Jeffrey Pang, and Shobha Venkataraman. 2011. Identifying diverse usage behaviors of smartphone apps. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. ACM, 329–344.

[50] Carl Yang, Xiaolin Shi, Luo Jie, and Jiawei Han. 2018. I Know You'll Be Back: Interpretable New User Clustering and Churn Prediction on a Mobile Social Application. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 914–922.

[51] Fengpeng Yuan, Xianyi Gao, and Janne Lindqvist. 2017. How busy are you?: Predicting the interruptibility intensity of mobile users. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 5346–5360.

[52] Sha Zhao, Julian Ramos, Jianrong Tao, Ziwen Jiang, Shijian Li, Zhaohui Wu, Gang Pan, and Anind K Dey. 2016. Discovering different kinds of smartphone users through their application usage behaviors. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 498–509.