ALOE: Active Learning based Opportunistic Experience Sampling for Smartphone Keyboard driven Emotion Self-report Collection

Surjya Ghosh*, Bivas Mitra†, Pradipta De‡
*Department of Computer Science and Information Systems, BITS Pilani Goa, INDIA
†Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, INDIA
‡Microsoft Corporation, USA
Email: surjyag@goa.bits-pilani.ac.in, bivas@cse.iitkgp.ac.in, prade@microsoft.com

2022 10th International Conference on Affective Computing and Intelligent Interaction (ACII) | 978-1-6654-5908-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ACII55700.2022.9953819

***Abstract*—Smartphone keyboard interaction based emotion detection systems are used widely to provide value-added services such as mental health monitoring, keyboard layout optimization, guided response generation. At the core of these services lie a machine learning model, which automatically infers emotion based on keyboard interaction pattern. To train these models, the emotion ground truth labels are typically collected as emotion self-report by conducting an Experience Sampling Method (ESM) based study. However, as responding to repetitive self-report probes is time-consuming and fatigue-inducing, efficient self-report collection approaches are essential that avoid probing at inopportune moments and reduce survey fatigue. To address this problem, we propose an active learning based framework, ALOE (Active Learning based Opportunistic Experience Sampling for Emotion Self-report Collection) that automatically decides to avoid probing at the unfavorable moments based on the typing signatures captured from smartphone keyboard interaction sessions. We bootstrap the framework with a few labeled instances (typing session) and allow the learner to probe (or query) the user only when it is least confident about an instance (typing session) and retrain accordingly. This way, we reduce the number of probes required (and therefore user engagement) and yet probe at the opportune moments. We evaluate ALOE in a 3-week in-the-wild study involving 18 participants, who record their smartphone keyboard interaction patterns and emotion self-reports during this period. The experimental results demonstrate that ALOE requires 56% less inopportune self-reports to train the probing moment detection learning model and yet detects the probing moments accurately with an average F-score of 93%.**

***Index Terms*—Emotion self-report, Active learning, Experience Sampling Method, User engagement, Survey fatigue**

I. Introduction

Smartphone typing-based emotion detection systems have shown potential for different applications such as unobtrusive mental health monitoring [1]–[4], interface design [5], [6], guided response generation [7], auto-suggestion usage optimization [8]. At the core of these systems, there is a machine learning model for automatic emotion inference. To train these models, different keyboard interaction characteristics are correlated with the emotion ground truth labels collected as manual self-reports from the Experience Sampling Method (ESM) [9], [10] driven study. But as manual self-reporting is time-consuming, fatigue-inducing, and burdensome, efficient self-report collection approaches to probe at opportune moments are required so that the survey fatigue and self-reporting effort can be reduced.

In the existing literature, many smartphone based interruptibility-aware notification strategies are developed to probe at the opportune moments [11], [12]. For example, Ghosh et al. developed a 2-phase model driven ESM schedule to optimize the probing rate and self-report timeliness [13]. Fischer et al. showed that participants react faster to probes when they are delivered immediately after completing a task on mobile (e.g., reading a text message) [14]. In [15], authors demonstrated that last survey response, phone's ringer mode can be leveraged to identify suitable probing moments. Specifically, in case of smartphone keyboard interaction, the authors extracted a set of time-domain and frequency-domain features to demonstrate their utility in opportune probing moment detection [16]. However, to the best of our knowledge, most of the prior works overlooked the significance of such signatures while developing an opportunistic probing strategy using *as few as possible* self-reported instances.

The key requirement to develop an opportunistic probing strategy using few self-reports is to establish that there is a distinguishing pattern in the keyboard interaction between opportune and inopportune probing moments. In machine learning literature, active learning has been used effectively to reduce the need for a large amount of annotated data [17], [18]. To develop the model, an active learning strategy requires a few annotated samples to begin with and later it can automatically decide the important instances to query for and fine-tune the model [17]. This helps the model to perform the desired task with relatively fewer labels. The same approach can be used for keyboard-based emotion self-report collection. We envision that (a) developing a base model with a few self-reports, and (b) gradually improving it based on the newly acquired self-reports (only when required) can identify the opportune probing moments accurately and with fewer self-report requests (and therefore less user engagement).

We, in this paper, propose an active learning based framework ALOE for Experience Sampling Method driven emotion self-report collection for smartphone keyboard interaction

(Section V). The proposed method aims to identify the suitable probing moments for a typing session[1], based on the keyboard interaction patterns (not textual content) in that session. ALOE learns from a few seed sessions (used for initialization) that there is an observable difference in the keyboard interaction parameters (e.g., session length, session duration, error rate) and previous ESM response between inopportune and opportune probing moments. It leverages these characteristics to decide when to trigger the probe to the user for emotion self-report (Section IV). Once the user completes typing in a session, ALOE decides whether the current moment is opportune for probing. ALOE adopts the least confidence query strategy to probe the user for the emotion self-report, where the model probes the user only when it is uncertain about its current probing decision, and skips probing when the model confidently decides the current moment as inopportune. This not only allows the model to reduce the number of probes to be responded by the user, but also ensures that probing is done at the opportune moments only. Along the process, whenever a new self-report is obtained from the user, ALOE retrains the model to make it more accurate in detecting the opportune probing moments.

We evaluated ALOE in a 3-week in-the-wild study involving 18 participants (Section VI). We developed an Android application and used it as experiment apparatus during the study (Section III). The app encompasses a QWERTY keyboard for tracking typing interactions (not text) and a probing UI for emotion self-report collection. The interface allows the user to record one of the four emotions (*happy, sad, stressed, relaxed*) after completing a typing session in any application. Additionally, the self-report UI allows the user to indicate (via *No Response* label) whether the current probing moment is inopportune. During the study, we collected ≈ 2500 labeled typing sessions, which reveal that typing characteristics (e.g., session length, session duration, error rate, typing speed) and previous self-report vary between inopportune and opportune probing moments. The active learner (in the ALOE framework) learns these characteristics from a few seed samples (Section IV-C) and by querying self-reports for the uncertain sessions retrains itself to detect the opportune moments. It reduces the inopportune probe self-reports by 56% to train the learner for probing moment detection (Section VI-B), while detects the opportune probing moments with an average F-score of 93% (Section VI-C).

II. Related Works

The Experience Sampling Method (ESM) is a widely used tool in psychology and behavioral research for in-situ sampling of human behavior, thoughts, and feelings [9], [10]. In this section, we discuss the related literature on interruptibility-aware mobile-based ESM strategies, and highlight their shortcomings. We also introduce the basic concept of active learning and its application in different contexts to reduce the annotation effort.

[1]Session is defined as the time spent at-a-stretch on a single application.

A. Interruptibility-aware Mobile-based ESM Design

Many emotion detection applications have used smartphone based emotion self-reports collection strategy as smartphones offer the flexibility of collecting rich contextual data during in-situ sampling [19], [20]. The advancements in this field demonstrate that different contextual information (e.g., location, app usage) can be leveraged to probe at the opportune moments [21], [22]. While the usage of additional contextual information is effective in designing the interruptibility-aware ESM approach, these information may not be used in an emotion self-report collection study due to other limitations such as privacy concerns [23]. As a result, researchers need to use only the study specific data during the probing strategy formulation. Specifically, in case of smartphone keyboard based emotion detection, different keyboard interaction parameters (e.g., typing speed, error rate) are to be used to detect the opportune probing moments while probing a user for emotion self-report [13], [16].

B. Active Learning for Reducing Labeling Effort

The basic idea of active learning is that if a model is trained intelligently with informative instances, it can perform well even with less training data [17]. This idea is useful in different scenarios (e.g., image classification [24], [25], image retrieval [26], image captioning [27]), where obtaining labels is expensive (time-consuming, resource-consuming) [28]. Broadly, there are two types of active learning algorithms - (a) stream-based, where the unlabeled samples are generated as a stream of data [29]; (b) pool-based, where a large pool of unlabeled samples is available [30]. The active learner is initialized with a set of labeled samples (known as seed samples), and then it is allowed to select the next unlabeled instance and decide, whether it should query for the label of that instance. To select the next unlabeled instance, different query strategies are used [31]–[33]. Among these, uncertainty sampling is the most widely used which queries for the instance the model is most uncertain about [18], [30], [31], [34]. In existing literature, different notions of uncertainly are used, e.g. margin [34], least confidence [18], entropy [31].

Summarizing the discussion of the related literature, we note that while developing an opportunistic probing strategy based on smartphone keyboard interaction, different interaction parameters (e.g., typing speed, typing mistake) can be utilized. At the same time, to reduce the number of probes at the unfavorable times, active learning can be applied. This suggests instantiating the opportune moment detection model with few labels (self-reports) and then retrain the model once additional labels (self-reports) for informative instances are collected. This approach not only helps to reduce the user engagement (in terms of self-reports), but also helps to probe at the opportune moments; which is the objective of this work.

III. Field Study

A. Experiment Apparatus

We have designed the keyboard app (Fig. 1a) based on Android Input Method Editor (IME) facility. It is same as

QWERTY keyboard with additional capability of tracing user's typing interactions. We do not store any alphanumeric character because of privacy reason. We have also obtained the IRB approval prior to data collection.

Tracing Keyboard Interactions: We define *session* as the time period spent by the user at-a-stretch on a single application. We record the timestamp of every touch event within a session and compute the interval between two consecutive touch events as *Inter-tap duration (ITD)*. For instance, we represent a session S of length $S_l(=n)$ as a sequence of timestamps $[t_1, t_2, t_3, ...t_n]$, depicting the respective touch events, with session duration $S_d = t_n - t_1$. We measure ITD as $v_i = t_{i+1} - t_i$, which reflects the typing speed of the user; higher value of ITD indicates lower typing speed. Hence, a session S may be further expressed as a sequence of ITDs, $S = [v_1, v_2, v_3, ..., v_n]$, where v_i indicates the i^{th} ITD. Additionally, we record the usage of the backspace or delete keys pressed in a session, which helps to identify the amount of typing mistakes made in a session.

Collecting Emotion Self-reports & Labeling Probing Moments: We also collect self-reported emotions from users. Once user completes typing in an application and switches from the current application, we probe her for the emotion self-report (*happy, sad, stressed, relaxed*) as shown in Fig. 1b. We select these emotions based on the Circumplex model (Fig. 1c) of emotion [35], as they represent largely represented emotion from separate quadrants, which makes self-reporting easier for the user. We keep the interface simple by explicitly recording emotion and do not consider the intensity of perceived emotion, which can make self-reporting difficult. We also keep the provision of *No Response*, so that user can skip self-reporting by selecting this option. Whenever the user reports *No Response*, the probing moment is considered inopportune, while any emotion (*happy, sad, stressed, relaxed*) response is considered opportune.

B. Study Procedure

We recruited 22 participants (18M, 4F) aged between 20 to 35 years from our university. We installed the application on their smartphones and asked them to use it for 3 weeks for regular typing activities and emotion self-reporting. We also informed that once they complete typing in an application and change it, they may receive a self-report pop-up, where they have to record their perceived emotion. They were further instructed that if the probe appears at an inopportune moment and they want to skip responding, they should select the *No Response* button instead of dismissing the pop-up. However, it was observed that 4 participants did not provide sufficient self-reports ($\leq$50), so we dropped these users and performed the analysis on the remaining 18 (15M, 3F) participants. We obtained the IRB approval before initiating the user study.

IV. Data Analysis: Key Findings

We have collected a total of 2540 sessions. The average number of sessions per user is 141.1 (std. dev 122.4). We have recorded 1573 (61.9%) sessions as opportune and 967 (38.1%) sessions as inopportune (Fig. 2). This finding demonstrates that a large number of sessions are recorded as inopportune. Therefore, availability of a mechanism to automatically identify these moments based on the *interaction pattern* and avoid probing at the inopportune moments can significantly reduce the number of probes. Next, we discuss the interaction characteristics to detect the probing moments.

A. Typing Features for Probing Moment Detection

Guided by earlier works [13], [16], we extract a set of typing features of a session S as (a) typing speed (S_{MSI}), (b) error rate (S_{Er}), (c) session length (S_l), (d) session duration (S_d) to characterize the difference between opportune and inopportune probing moments. We represent the typing speed in a session S as Mean Session ITD (MSI), where we compute the mean of all ITDs present in session S as $S_{MSI} = \frac{\sum_{i=1}^{n-1} v_i}{n-1}$. We also compute the typing mistakes performed in a session by counting the total number of backspace (or delete) key pressed in a session (say, c), and compute as $S_{Er} = \frac{c}{n}$. To handle the inter-subject variability [36], we normalize each feature as $x' = \frac{x - min(X)}{max(X) - min(X)}$, where $X \in \{S_{MSI}, S_{Er}, S_l, S_d\}$ is the set of values recorded for a feature across all individuals, x is one instance of the set X, $min(X)$, $max(X)$ indicate minimum and maximum of the set X.

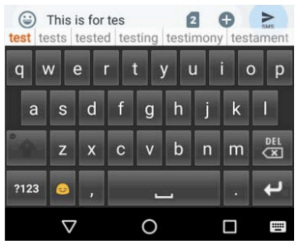
B. Previous Response Feature for Probing Moment Detection

We also use one self-reporting characteristic to distinguish between opportune and inopportune probing moments. Earlier works suggest that previous ESM response can be used as a good indicator of current ESM response [13]. Accordingly, we use the self-report associated with $(n-1)^{th}$ typing session as a feature to determine the response for n^{th} session. We use this feature as a binary one, if the self-report is one of the four emotions (*happy, sad, stressed, relaxed*) the value is set to 0, otherwise (for *No Response*) the value is set to 1.

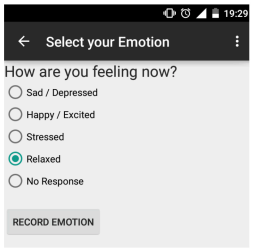
C. Feasibility Analysis for Reducing Inopportune Probes

We aim to develop a machine learning model that (a) detects the opportune probing moments based on keyboard interaction patterns, and (b) uses as few as possible training samples from inopportune moments so that the user needs to respond to fewer self-reports (and therefore the survey fatigue is reduced).

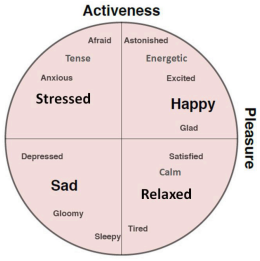
To investigate this, we analyze the collected dataset further in terms of underlying pattern of the feature values. Since we represent every data point (session) in terms of four typing based features and one emotion response feature, first we reduce the dimensionality of the data to visualize it in a 2-D plane. We apply PCA (principal component analysis) [37] on the collected dataset (by setting number of principal components to two) and show the outcome in a scatterplot in Fig. 3. The figure reveals that for many of the inopportune moment samples the value of first principal component (PC1) is relatively large. For example, it is observed that $\approx 30\%$ of the inopportune data points have PC1 value greater than or equal to 100 (and no opportune moment sample has a value greater or equal to 100). This points to the fact that,



(a) App keyboard



(b) Self-reporting UI



(c) Circumplex model [35]

Fig. 1: Experiment Apparatus - (a) The app keyboard was used to trace typing interactions, (b) the self-report UI was used to collect the emotion self-report, (c) the UI was designed guided by the Circumplex model of emotion

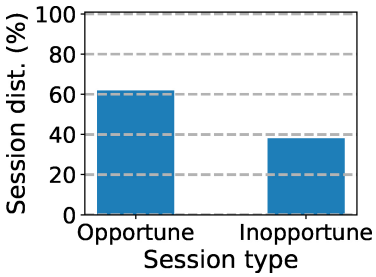


Fig. 2: Distribution of inopportune and opportune sessions

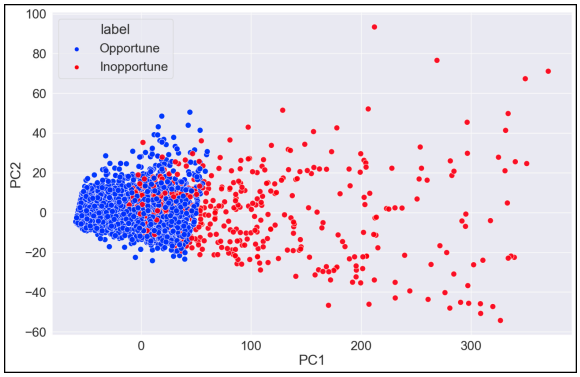


Fig. 3: The visualization reveals a noticeable difference between the inopportune and opportune probing moments, which may be leveraged by a machine learning model to reduce the requirement of large number of inopportune data points.

to discriminate the opportune and inopportune moments, the machine learning model may not need a large volume of inopportune labels from this region. This helps to significantly reduce the number of inopportune data points required to train the machine learning model.

In summary, we observe that although the participants reported a large number of inopportune probes, there is a noticeable difference between the opportune and inopportune probing moments based on the identified features. Therefore, it may be possible to learn this difference using a machine learning model with relatively few number of inopportune data points, which motivates us to develop the active learning framework as described next.

V. ALOE: Active Learning based Opportunistic Experience Sampling Framework

In this section, we discuss the ALOE framework (Fig. 4). In the first step of the framework, we use a few typing sessions (marked as opportune or inopportune) as seeds to train a machine learning model for detecting the opportune probing moments. This is the base model of the proposed framework. At the end of the base model construction (using the seed sessions), new typing sessions keep on getting generated from users' typing activities. These typing sessions are generated as streams and it is to be decided whether the probing moments for these typing sessions are opportune or not. To decide that, each of the typing sessions (as generated) is sent to the base model. The base model returns its confidence about detecting the current session as opportune or not. If the model responds with a high confidence value that the current session is inopportune, we do not ask the user for a self-report. Otherwise, we probe the user for the self-report and retrain the base model with the newly obtained user input. This way we decide the probing moment for every typing session and retrain the base model as often as required. This strategy not only allows to avoid probing at the inopportune moments, but also improves the learner in detecting opportune probing moments by retraining. We discuss each of these steps in detail next.

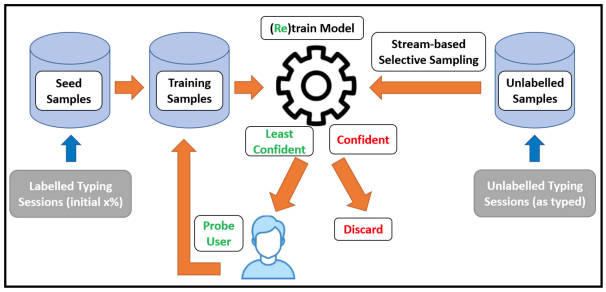


Fig. 4: The architecture of the ALOE framework. First, a set of typing sessions (marked as opportune or inopportune) are identified and used as seed to train the model of the framework. After that, as new typing sessions are generated as stream, those are sent to the model to decide if the session is opportune or not. If the model is confident that the current session is inopportune, the user is not probed; otherwise the user is probed. Whenever a new response is collected from the user, the model is retrained to make it more accurate in detecting the opportune probing moments.

Seed Sample Identification: We use a set of typing sessions (marked as opportune or inopportune) as seed samples. In this case, as typing sessions are generated with time, based on users' typing activities, we do not have the entire dataset available from the beginning. Therefore, we decide to accumulate initial x% (we set the value of x in the experimental setup, section VI-A2) typing sessions from each of the users to gather the seed samples.

Base Model Construction: ALOE consists of a machine learning model that decides whether the probing moment for a typing session is opportune or not. To train the model, we use the identified seed samples. We extract the same set of features from every typing session as mentioned earlier (section IV-A, IV-B). We implement Random Forest to train the model using 100 decision trees with setting the maximum depth of the tree as unlimited, both of these (a large number of trees, maximum depth) help to counter overfitting. After training the model with seed samples, we have the base model of the framework ready for deciding the probing moment for a typing session.

Opportunistic Probe Triggering Decision: Once the base model is trained with the seed typing sessions, the subsequent typing sessions need to be marked as opportune (or not) with minimal user involvement. As typing sessions are generated like a stream, we apply selective sampling [38] on this data stream to decide, for which typing session the user needs to be probed for emotion self-report. In specific, every newly generated typing session is sent to the base model to find its confidence in identifying the current session as opportune (or not). We adopt Least Confidence (a category of Posterior probability-based Strategies) [39] strategy while probing the user for a typing session. In this strategy, we probe the user for a typing session if the probability of detecting that typing session as inopportune is less than 0.5. We select 0.5 as the threshold as we have two classes only (inopportune and opportune). Notably, if the probability of detecting inopportune moment is very small (i.e., the probing moment is likely to be opportune), the user is probed. We want to collect samples anyway at these moments because the moments are opportune (therefore suitable for probing). This way we could avoid probing the users in some of the inopportune moments (when the model is confident) and reduce the number of probes to be attended by the user.

Model Retraining: We retrain the base model on every occasion, whenever a response is collected from the user for a typing session. This allows improving the learner in detecting the opportune probing moments accurately. The probing and retraining are stopped once the required number of typing sessions are marked. At the end of this phase, we have a machine learning model capable of identifying the probing moment as opportune (or not) based on typing interactions.

VI. Evaluation

In this section, we discuss the experimental evaluation of ALOE. First, we describe the experiment setup, which includes the description of the baselines, evaluation strategy, and the performance metrics. Later, we analyze ALOE's performance in reducing inopportune probes and detecting opportune probing moments. We also discuss the influence of seed samples, influence of retraining, and the explainability of ALOE.

A. Experiment Setup

1) Baselines: We compare the performance of ALOE with the following baseline models as proposed by Ghosh et. al [16]. The authors proposed to combine different time-domain and frequency-domain characteristics extracted from typing sessions to determine the opportune probing moments.

- **Time-domain (TD):** This model extracts a set of time-domain features (session length, session duration, average session speed, error rate) from typing sessions to train a Random Forest based model for detecting the opportune moments.
- **Frequency-domain (FD):** In this model, the authors proposed to transform the typing session details to frequency domain first applying Discrete Fourier Transform (DFT) [40]. Then, the transformed values are passed through a filter to purge the complex components. Later, a peak detection algorithm is applied to identify the top-3 amplitudes. The top-3 amplitudes and the number of peaks are used as features to train a Random Forest model to detect the opportune probing moments.
- **Combined (Comb):** This model consists of both the time-domain and frequency-domain features. It also implements a Random Forest based model.

2) Evaluation Strategy: To train and evaluate ALOE, we split the dataset into 3 parts - (a) seed samples, (b) opportunistic query samples, and (c) test samples. To train the base model of the active learner in ALOE, we combine the first 20% typing sessions from every user. The next 60% typing sessions for each user are used opportunistically to query user for additional self-report (this allows to reduce the number of probes the users need to respond). Whenever a new self-report (inopportune or opportune) for a typing session is acquired the base model is retrained. At the end of this 60% typing sessions, we have the fully trained model for the user, which is evaluated using the remaining 20% typing sessions of the user. These steps are repeated for every user.

In order to train the baselines, we have used initial 80% typing sessions (marked as opportune or inopportune) of each user. This segment of data for every user is combined to form the training set. However, for evaluation, we evaluate every user independently. The final (left out) 20% typing sessions of every user is treated as a testing set (for the specific user).

3) Performance Metrics: We use the following metrics to evaluate ALOE,

F-score: We use F-score as the metric to decide the opportune probing moment detection performance. First, we compute the user-wise F-scores, which are averaged over all users to report the performance of ALOE.

Inopportune Probe Reduction: To compute the inopportune probe reduction in ALOE, we find out how many fewer inopportune probes are answered by the users in comparison to the baselines. The baselines use self-reports from 80% sessions for training, while ALOE uses self-reports from initial 20% (as seed) and next 60% opportunistically. Therefore, the reduction stems from answering fewer inopportune probes from the opportunistic query samples. Specifically, in the 80% samples (20% seed, and 60% opportunistic query samples), if n_{ALOE}, n_{bl} are the number of inopportune probes answered by the users in ALOE and the baselines respectively, the reduction is computed as $\frac{(n_{bl}-n_{ALOE})*100}{n_{bl}}$.

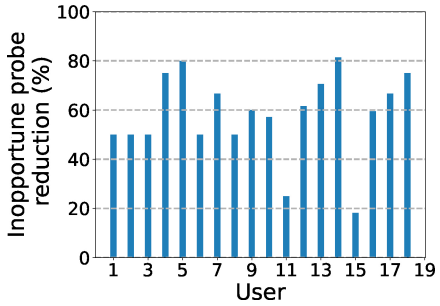
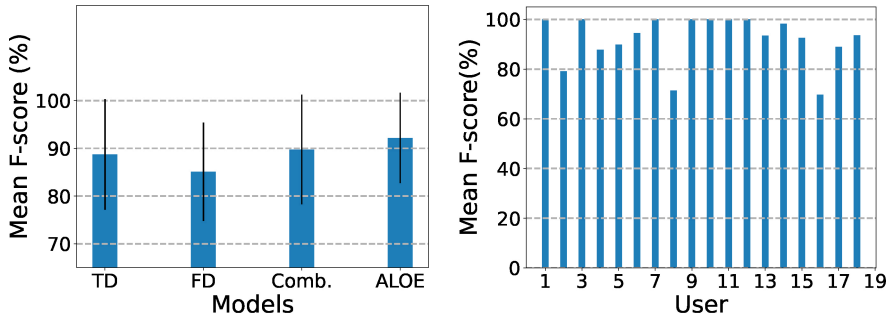


Fig. 5: ALOE's inopportune probe reduction performance for every user. For 83% of the users, there is a reduction of at least 50%, leading to an average reduction of 56%.



(a) Model-wise F-score

(b) User-wise F-score

Fig. 6: ALOE's opportune moment detection performance - (a) comparison with baseline F-score. Error bar indicates std. dev. (b) user-wise F-score

B. Performance Analysis: Inopportune Probe Reduction

In this section, we analyze the performance of ALOE in reducing the inopportune probes. We show the inopportune probe reduction using the ALOE framework in Fig. 5. It is observed that 55% of the users have a reduction of at least 60% and 83% of the users have a reduction of at least 50%. At the same time, for a few users (11, 15), the inopportune probe reduction is relatively less ($\leq$ 25%). Overall, we obtain an average reduction of 56.8% (std dev. 18%). But the important question, whether the inopportune probe reduction influences the probing moment detection performance, is discussed next.

C. Performance Analysis: Probing Moment Detection

We show ALOE's performance in detecting the opportune probing moments in Fig. 6. First, we compare the performance of ALOE with the baselines in Fig. 5a. ALOE returns a average F-score 93% (std. dev: 7.9%) outperforming all the baselines (TD: 88.1%, FD: 85.2%, Comb: 89%). We also report the user-wise F-score in Fig. 5b. It is observed that for all but 2 users (2, 8), the F-score is more than 80% leading to an average F-score of 93%. Notably, ALOE achieves these superior results despite being (re)trained with the fewer inopportune labels in comparison to the baselines.

D. Performance Analysis: Influence of Seed Samples

In this section, we analyze the performance of ALOE both in terms of probing moment detection and inopportune probe reduction with increasing number of seed samples (Fig. 7). We observe that with increasing number of seed samples, there is not much variation in probing detection accuracy. We envision this happens because the difference between inopportune and opportune probing moments (as shown in Fig. 3) can be easily identified using a few seed samples only and therefore increasing seed samples does not influence model performance much. On the contrary, with increase in the number of seed samples, the inopportune probe reduction rate drops (as the user needs to respond more number of self-reports during the seed collection phase). Therefore, it may be prudent to instantiate the model with approximately 15% to 20% seed sessions to obtain a high probing moment detection performance and a high reduction in inopportune probing rate.

E. Performance Analysis: Influence of Retraining

We also evaluate the influence of retraining on opportune probing moment detection performance. To achieve this, we keep the seed samples fixed (20%) and vary the number of opportunistic query samples. With the increasing number of query samples, the model gets more opportunity of retraining. We show the probing moment detection performance with increasing opportunistic query samples in Fig. 8. It is observed that with increase in the query sample, the probing moment detection performance improves gradually. It is also noted with increasing amount of the query sample, the standard deviation in user-wise F-score reduces indicating that variation of F-scores for different users diminishes. This highlights that using the base model (trained only with seed samples) does not return high probing moment detection performance; retraining it as and when required based on the opportunistic query samples improves the probing moment detection performance.

F. Performance Analysis: Influence of Threshold

We assess the impact of the variation of the probing threshold on probing moment detection and inopportune probe reduction in Fig. 9. It is observed that when the threshold is less (i.e., the probing criteria is conservative), we end up probing fewer times resulting in higher reduction in inopportune probing. But this reduces the probing moment detection performance (90%). However, as we increase the threshold (i.e., the probing criteria becomes relaxed), we end up probing more (resulting in comparatively less saving in probe reduction). But this provides the learner to gather more self-reports and retrain accordingly. As a result, the probing moment detection performance improves. However, beyond a certain value of threshold ($\geq$ 0.4), the probing moment detection performance does not improve, but the probing reduction drops. Therefore, a threshold in the range of 0.4 to 0.5 can be used.

G. ALOE Framework Explaianability

We perform the explainability analysis of the proposed model in ALOE using SHAP (SHapley Additive exPlana-

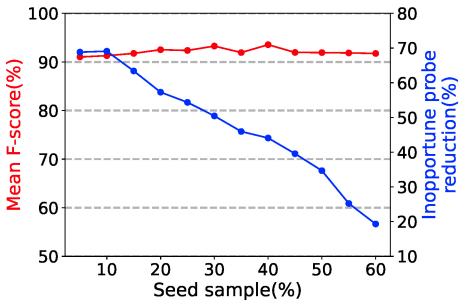


Fig. 7: Variation in probing moment detection performance and inopportune probe reduction with different amount of seed sample to train the base model.

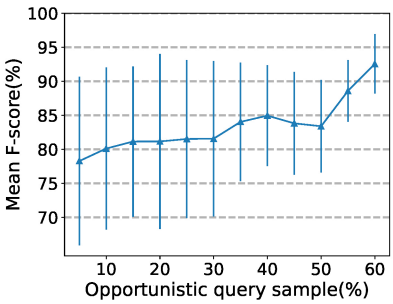


Fig. 8: Variation in probing moment detection performance with different amount of opportunistic query samples to retrain the base model.

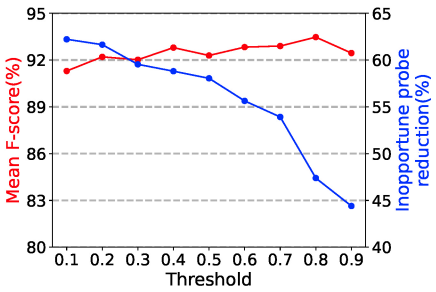


Fig. 9: Variation in probing moment detection performance and inopportune probe reduction with different thresholds.

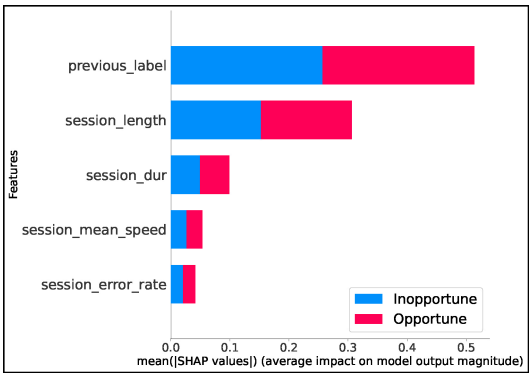


Fig. 10: Explainability analysis using SHAP reveals that previous response and session length are the top two features in detecting opportune probing moments.

tions) [41], where Shapley index of a model feature exhibits its contribution to determine the predicted class for an instance [42]. In Fig. 10, we compute the shapley index for each of the features on the test set and show the mean absolute SHAP values for each feature. This analysis reveals that the previous response has the strongest influence in classifying the opportune probing moments followed by the session length. Session duration is found to have a moderate impact, while session typing speed and session error rate have minimal influence on the model performance.

VII. Discussion and Future Works

The findings from initial explorations of ALOE are promising. However, we envision following aspects need to be considered for applying ALOE in an online setting or to extend it for other ESM studies.

Deployment Considerations: To deploy ALOE in online setting, we need to figure out (a) the number of seed samples and (b) the retraining frequency. We recommend the initial 20% to 30% of self-reports (acquired from a self-report collection study) can be used as the seed to initialize the model, as it provides good probing moment detection performance and high probing reduction rate (Fig. 7). The frequency of retraining can be decided based on the data volume or when the distribution of stream data changes significantly than training data to overcome the retraining overhead [43], [44].

Reducing Seed Selection Overhead: In the recent past, zero-shot learning, one-shot learning, and few-shot learning paradigms are used effectively to tackle the challenge of annotating one or a few instances for the classification task [45], [46]. We also aim to borrow concepts from these approaches so that the manual self-report requirement of seed samples can be reduced significantly in our future work.

Generalizing ALOE for other ESM studies: The crux of ALOE lies in figuring the difference between opportune and inopportune probing moments. The investigator, who is designing the study may be able to identify the parameters (features) leading to suitable probing moments. Once these factors are identified and the learner is trained with the seed samples, the learner should be able to identify the informative samples and retrain. So, we believe with initial guidance from the experimenter the proposed framework can be extended to other ESM-driven studies also.

VIII. Conclusion

In this paper, we propose an active learning-based framework ALOE for experience sampling, which opportunistically decides to probe users for emotion self-reports based on smartphone keyboard interaction patterns. The active learner embedded in the framework is instantiated with a few labeled typing sessions. Based on these sessions, the learner figures out the key differences between inopportune and opportune sessions in terms of typing session length, session duration, error rate, typing speed, and previous self-report. Leveraging these, it decides to query (or probe) the user based on the least confidence strategy (only for the uncertain sessions) in a selective sampling approach for the stream of typing sessions produced from the user's typing activities. This approach allows the learner to learn from more informative instances and discard probing for the confident ones. Thus, it provides an opportunity to reduce the number of probes yet ensures that the probing is performed at an opportune moment only. The validation of the ALOE framework with a 3-week in-the-wild study involving 18 participants reveals that it needs 56% fewer inopportune self-report probes to train the model and yet detects the opportune probing moments with an average F-score of 93%.

REFERENCES

[1] M. Ciman and K. Wac, "Individuals' stress assessment using human-smartphone interaction analysis," *IEEE Transactions on Affective Computing*, vol. 9, no. 1, pp. 51–65, 2016.

[2] S. Ghosh, N. Ganguly, B. Mitra, and P. De, "Evaluating effectiveness of smartphone typing as an indicator of user emotion," in *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 2017, pp. 146–151.

[3] R. Wampfler, S. Klingler, B. Solenthaler, V. R. Schinazi, and M. Gross, "Affective state prediction based on semi-supervised learning from smartphone touch data," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–13.

[4] S. Ghosh, N. Ganguly, B. Mitra, and P. De, "Tapsense: Combining self-report patterns and typing characteristics for smartphone based emotion detection," in *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*, 2017, pp. 1–12.

[5] N. Bin Hannan, K. Tearo, J. Malloch, and D. Reilly, "Once more, with feeling: Expressing emotional intensity in touchscreen gestures," in *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, 2017, pp. 427–437.

[6] F. Noori and M. Kazemifard, "Aubue: An adaptive user-interface based on users' emotions," *Journal of Computing and Security*, vol. 3, no. 2, pp. 127–145, 2016.

[7] C.-Y. Huang, T. Labetoulle, T.-H. K. Huang, Y.-P. Chen, H.-C. Chen, V. Srivastava, and L.-W. Ku, "Moodswipe: A soft keyboard that suggests messages based on user-specified emotions," *arXiv preprint arXiv:1707.07191*, 2017.

[8] S. Ghosh, K. Hiware, N. Ganguly, B. Mitra, and P. De, "Does emotion influence the use of auto-suggest during smartphone typing?" in *Proceedings of the 24th International Conference on Intelligent User Interfaces*, 2019, pp. 144–149.

[9] R. Larson and M. Csikszentmihalyi, "The experience sampling method." *New Directions for Methodology of Social & Behavioral Science*, 1983.

[10] J. M. Hektner, J. A. Schmidt, and M. Csikszentmihalyi, *Experience sampling method: Measuring the quality of everyday life*. Sage, 2007.

[11] N. V. Berkel, D. Ferreira, and V. Kostakos, "The experience sampling method on mobile devices," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, p. 93, 2017.

[12] M. Pielot, B. Cardoso, K. Katevas, J. Serrà, A. Matic, and N. Oliver, "Beyond interruptibility: Predicting opportune moments to engage mobile phone users," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, pp. 1–25, 2017.

[13] S. Ghosh, N. Ganguly, B. Mitra, and P. De, "Designing an experience sampling method for smartphone based emotion detection," *IEEE Transactions on Affective Computing*, 2019.

[14] J. E. Fischer, C. Greenhalgh, and S. Benford, "Investigating episodes of mobile phone activity as indicators of opportune moments to deliver notifications," in *Proceedings of ACM MobileHCI*, 2011, pp. 181–190.

[15] M. Pielot, R. de Oliveira, H. Kwak, and N. Oliver, "Didn't you see my message?: predicting attentiveness to mobile instant messages," in *Proceedings of the ACM SIGCHI*, 2014, pp. 3319–3328.

[16] S. Ghosh, S. Mandi, B. Mitra, and P. De, "Exploring smartphone keyboard interactions for experience sampling method driven probe generation," in *26th International Conference on Intelligent User Interfaces (ACM IUI)*, 2021, pp. 133–138.

[17] B. Settles, "Active learning literature survey," *Technical Report, University of Wisconsin-Madison Department of Computer Sciences*, 2009.

[18] A. Culotta and A. McCallum, "Reducing labeling effort for structured prediction tasks," in *AAAI*, vol. 5, 2005, pp. 746–751.

[19] M. Pielot, T. Dingler, J. S. Pedro, and N. Oliver, "When attention is not scarce-detecting boredom from mobile phone usage," in *Proceedings of the ACM UbiComp*, 2015, pp. 825–836.

[20] V. Pejovic, N. Lathia, C. Mascolo, and M. Musolesi, "Mobile-based experience sampling for behaviour research," in *Emotions and Personality in Personalized Services*. Springer, 2016, pp. 141–161.

[21] A. Mathur, N. D. Lane, and F. Kawsar, "Engagement-aware computing: Modelling user engagement from mobile contexts," in *Proceedings of ACM UbiComp*, 2016.

[22] M. Pielot, B. Cardoso, K. Katevas, J. Serrà, A. Matic, and N. Oliver, "Beyond interruptibility: Predicting opportune moments to engage mobile phone users," *Proceedings of the ACM IMWUT*, vol. 1, no. 3, 2017.

[23] A. Raij, A. Ghosh, S. Kumar, and M. Srivastava, "Privacy risks emerging from the adoption of innocuous wearable sensors in the mobile environment," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2011, pp. 11–20.

[24] W. Fu, M. Wang, S. Hao, and X. Wu, "Scalable active learning by approximated error reduction," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1396–1405.

[25] J. Choi, K. M. Yi, J. Kim, J. Choo, B. Kim, J. Chang, Y. Gwon, and H. J. Chang, "Vab-al: Incorporating class imbalance and difficulty with variational bayes for active learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[26] B. Barz, C. Käding, and J. Denzler, "Information-theoretic active learning for content-based image retrieval," in *German Conference on Pattern Recognition*. Springer, 2018, pp. 650–666.

[27] Y. Deng, K. Chen, Y. Shen, and H. Jin, "Adversarial active learning for sequences labeling and generation." in *IJCAI*, 2018, pp. 4012–4018.

[28] X. J. Zhu, "Semi-supervised learning literature survey," *Technical Report, University of Wisconsin-Madison Department of Computer Sciences*, 2005.

[29] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Machine learning*, vol. 15, no. 2, pp. 201–221, 1994.

[30] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *SIGIR'94*. Springer, 1994, pp. 3–12.

[31] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, 2008.

[32] S.-J. Huang, R. Jin, and Z.-H. Zhou, "Active learning by querying informative and representative examples," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 10, pp. 1936–1949, 2014.

[33] S. Dasgupta and D. Hsu, "Hierarchical sampling for active learning," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 208–215.

[34] M.-F. Balcan, A. Broder, and T. Zhang, "Margin based active learning," in *International Conference on Computational Learning Theory*. Springer, 2007, pp. 35–50.

[35] J. A. Russell, "A circumplex model of affect," *Journal of Personality and Social Psychology*, vol. 39, no. 6, pp. 1161–1178, 1980.

[36] R. Taib, J. Tederry, and B. Itzstein, "Quantifying driver frustration to improve road safety," in *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, 2014, pp. 1777–1782.

[37] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[38] O. Dekel, C. Gentile, and K. Sridharan, "Selective sampling and active learning from single and multiple teachers," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 2655–2697, 2012.

[39] D. Tuia, M. Volpi, L. Copa, M. Kanevski, and J. Munoz-Mari, "A survey of active learning algorithms for supervised remote sensing image classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 3, pp. 606–617, 2011.

[40] S. Winograd, "On computing the discrete fourier transform," *Mathematics of computation*, vol. 32, no. 141, pp. 175–199, 1978.

[41] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., 2017, pp. 4765–4774.

[42] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "From local explanations to global understanding with explainable ai for trees," *Nature Machine Intelligence*, vol. 2, no. 1, pp. 2522–5839, 2020.

[43] Z. Wan, X. Xia, D. Lo, and G. C. Murphy, "How does machine learning change software development practices?" *IEEE Transactions on Software Engineering*, 2019.

[44] S. Schelter, F. Biessmann, T. Januschowski, D. Salinas, S. Seufert, and G. Szarvas, "On challenges in machine learning model management," 2018.

[45] W. Wang, V. W. Zheng, H. Yu, and C. Miao, "A survey of zero-shot learning: Settings, methods, and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, 2019.

[46] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Computing Surveys (CSUR)*, vol. 53, no. 3, pp. 1–34, 2020.