Towards Efficient Emotion Self-report Collection Using Human-AI Collaboration: A Case Study on Smartphone Keyboard Interaction

PRAJWAL M*, Department of Electrical & Electronics Engineering, BITS Pilani Goa, India
AYUSH RAJ*, Department of Computer Science & Information Systems, BITS Pilani Goa, India
SOUGATA SEN, Department of Computer Science & Information Systems, BITS Pilani Goa, India
SNEHANSHU SAHA, APPCAIR, Department of Computer Science & Information Systems, BITS Pilani Goa, and HappyMonk AI Labs, India
SURJYA GHOSH, Department of Computer Science & Information Systems, BITS Pilani Goa, India

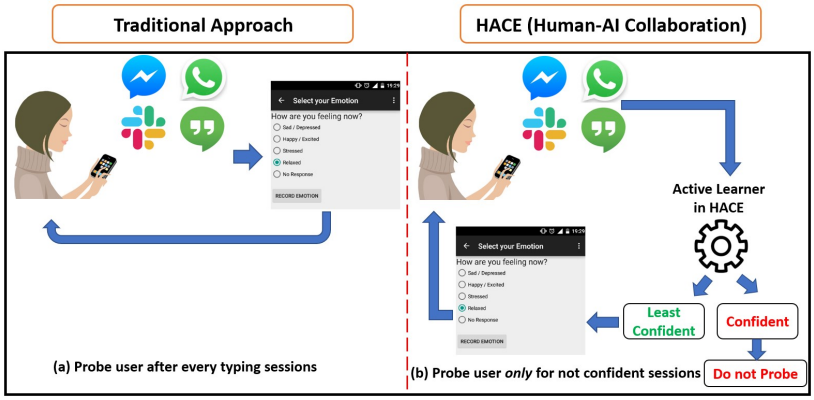


Fig. 1. Human-AI Collaborative Emotion Self-report Collection (HACE) framework for smartphone keyboard-based interaction scenario. (a) In traditional approach (absence of Human-AI collaboration), the user provides input for emotion self-report probe after *every* typing session. (b) In the HACE framework, a user is probed *only* for those typing sessions for which the self-report can't be estimated with high confidence. So, the user responds to fewer probes, and survey fatigue is reduced.

Emotion-aware services are increasingly used in different applications such as gaming, mental health tracking, video conferencing, and online tutoring. The core of such services is usually a machine learning model that automatically infers

*Both authors contributed equally to this research.

Authors' addresses: Prajwal M, f20180299g@alumni.bits-pilani.ac.in, Department of Electrical & Electronics Engineering, BITS Pilani Goa, India; Ayush Raj, f20180954@goa.bits-pilani.ac.in, Department of Computer Science & Information Systems, BITS Pilani Goa, India; Sougata Sen, sougatas@goa.bits-pilani.ac.in, Department of Computer Science & Information Systems, BITS Pilani Goa, India; Snehanshu Saha, snehanshus@goa.bits-pilani.ac.in, APPCAIR, Department of Computer Science & Information Systems, BITS Pilani Goa, and HappyMonk AI Labs, India; Surjya Ghosh, surjyag@goa.bits-pilani.ac.in, Department of Computer Science & Information Systems, BITS Pilani Goa, India.

2474-9567/2023/6-ART68 $15.00
https://doi.org/10.1145/3596269

its user's emotions based on different biological indicators (e.g., physiological signals and facial expressions). However, such machine learning models often require a large number of emotion annotations or ground truth labels, which are typically collected as manual self-reports by conducting long-term user studies, commonly known as Experience Sampling Method (ESM). Responding to repetitive ESM probes for self-reports is time-consuming and fatigue-inducing. The burden of repetitive self-report collection leads to users responding arbitrarily or dropping out from the studies, compromising the model performance. To counter this issue, we, in this paper, propose a Human-AI Collaborative Emotion self-report collection framework, *HACE*, that reduces the self-report collection effort significantly. HACE encompasses an active learner, bootstrapped with a few emotion self-reports (as seed samples), and enables the learner to query for *only* not-so-confident instances to retrain the learner to predict the emotion self-reports more efficiently. We evaluated the framework in a smartphone keyboard-based emotion self-report collection scenario by performing a 3-week in-the-wild study (N = 32). The evaluation of HACE on this dataset (≈11,000 typing sessions corresponding to more than 200 hours of typing data) demonstrates that it requires 46% fewer self-reports than the baselines to train the emotion self-report detection model and yet outperforms the baselines with an average self-report detection F-score of 85%. These findings demonstrate the possibility of adopting such a human-AI collaborative approach to reduce emotion self-report collection efforts.

CCS Concepts: • **Human-centered computing** → **Human computer interaction (HCI)**; • **Computing methodologies** → *Machine learning*; **Active learning settings**.

Additional Key Words and Phrases: Experience Sampling Method, ESM, Active learning, User engagement, Survey fatigue

ACM Reference Format:
Prajwal M, Ayush Raj, Sougata Sen, Snehanshu Saha, and Surjya Ghosh. 2023. Towards Efficient Emotion Self-report Collection Using Human-AI Collaboration: A Case Study on Smartphone Keyboard Interaction. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 2, Article 68 (June 2023), 23 pages. https://doi.org/10.1145/3596269

1 INTRODUCTION

Many emotion-aware applications, such as online meeting platforms and affective tutoring systems, have recently been designed to improve user experience and engagement [29, 48]. These applications typically use a machine learning (ML) model to automatically infer the user emotion and accordingly adapt the flow of the application to maintain user engagement. Generally, the ML models leverage one or more biological indicators such as facial expressions, speech patterns, and physiological signals to accurately infer the user's emotion [34, 39, 61]. However, one major challenge encountered while training these models is the requirement of a large number of emotion ground truth labels. The emotion ground truth labels are typically collected as emotion self-reports by running a long-term user study called Experience Sampling Method (ESM) [36]. But as ESM-driven studies require users to respond to repetitive emotion self-report questionnaires, the user burden increases significantly. Therefore, many users arbitrarily respond to the emotion self-reports or drop out in-between from the study [47, 50]. Both arbitrary self-reports and in-between dropouts impact the model quality and, therefore, the overall performance of these emotion-aware applications. Thus, research effort is required to develop efficient emotion self-report collection approaches to reduce the user burden.

In the existing literature, researchers have adopted different approaches to reduce the emotion self-report collection efforts in ESM studies. First, one of the most widely used approaches is to trigger the emotion self-report probes when a specific event occurs (e.g., after applying a particular stimulus (audio, video, image), or after performing a specific task (physical training, upon reaching the desired location)) instead of continuously probing users at a fixed time interval [11, 49]. While the event-triggered strategies help to reduce the number of probes compared to time-triggered ones, the complexity and number of probes can increase if the number of events to be monitored is large [5, 47]. Second, to counter these challenges, probing strategies have been designed to issue probes only when there is a significant variation in physiological signals (e.g., heart rate, GSR) [2]. Third, another line of work targets to reduce the self-report collection effort by probing at the opportune moments when the user is interruptible [28, 35, 70, 81]. These approaches typically leverage different contextual cues

(e.g., the transition from one activity to another [18], phone's ringer mode, and last survey response [51]) from sensor data to figure out the user's availability to respond to emotion self-report probes so that the interruption can be minimized [44–46]. Finally, applying the Human-AI collaborative approach is another commonly used practice to reduce annotation effort [43]. While such strategies are used in other domains (e.g., human activity recognition (HAR) [1]), to the best of our knowledge, no prior work investigated the application of the Human-AI collaborative approach for easing out the emotion self-report collection effort in ESM studies.

Developing a Human-AI Collaborative approach to reduce the emotion self-report collection effort requires addressing multiple challenges. First, it may not be feasible to devise a probe reduction strategy leveraging data from various sensors as a particular sensor stream may not be available for a given ESM study [71], or it may have privacy issues [54], or it may incur significant resource cost (e.g., GPS) [7]. Therefore, the devised approach should leverage *only* those data streams that are collected as part of the ongoing ESM study. Second, the selected sensor stream(s) must capture *noticeable* differences among different emotion self-reports so that the self-report probing strategies can be developed leveraging such differences. Finally, as the objective is to reduce the user burden (i.e., the number of emotion self-report probes), the Human-AI collaborative framework should figure out these differences confidently with little user intervention (i.e., it should require as few emotion self-reports as possible).

We, in this paper, propose a Human-AI Collaborative Emotion self-report collection framework, HACE, that reduces the emotion self-report collection effort in a long-term ESM study addressing the challenges mentioned above. The human and AI collaboration happens in making the emotion self-report collection (i.e., during the data collection and labeling stage of a human-in-the-loop approach [43]) more efficient. In specific, the framework encompasses an active learning strategy, which allows instantiating the learner with *a few* emotion self-reports (i.e., seed samples) and train a basic emotion self-report prediction model. Later, when more instances are available, they are passed through the emotion self-report model to predict the emotion self-report for those instances. If the learner is confident about the outcome, the predicted emotion self-report is used, and the user is not probed further. However, if the learner is not confident about the prediction, then a self-report probe is triggered to the user, and the learner is retrained with the newly (user) provided emotion self-report so that it becomes more accurate in predicting future emotion self-reports. In the entire process, as the emotion self-reports are collected from the user *only* when the learner is not confident, it requires the user to respond to fewer probes, and therefore, the user burden in the long-term ESM study is reduced substantially.

We demonstrate the working of the HACE framework with a case study (Section 3) on smartphone keyboard interaction. We selected the smartphone keyboard interaction for the case study for two reasons. First, it is one of the widely used modalities for emotion inference due to the overwhelming usage of different instant messaging applications [76]. Second, smartphones are the widely used device for the in-situ sampling of human behavior [5]. We present the schematic diagram of the HACE in the smartphone keyboard interaction scenario in Fig. 1. Unlike traditional approaches of seeking emotion self-report after every typing session (i.e., the time spent on a single app at-a-stretch before changing to the next one), we probe the user only for those sessions when the learner embedded in the HACE is not confident; thus helping to reduce the number of self-report probes to be responded by the users. For the case study, we developed an Android-based QWERTY keyboard that allows tracking typing interaction patterns (not the actual content) such as typing speed, typing duration, and typing error and collects emotion self-reports (*happy, sad, stressed, relaxed*) after the typing sessions. The participants used this app for their daily typing activities and emotion self-reporting.

We performed a 3-week in-the-wild user study involving 32 participants that led to the collection of ≈11,280 typing sessions (total duration of these sessions ≈203 hours). The analysis of this collected dataset reveals that typing characteristics (such as typing speed and session length) and previous responses vary significantly across four different emotion self-reports (Section 4.4). The active learner leverages these cues based on a few seed samples and trains a base model for emotion self-report prediction. The learner predicts the emotion self-report

once a typing session is generated, and if it is not confident about the outcome (i.e., can't predict the outcome with a high probability), it probes the user for the emotion self-report and retrains itself (Section 5). These Human-AI collaborative design choices help to reduce the self-report probes by 46% and yet detect the emotion self-reports with an average F-score of 85% (Section 6.2, 6.3). We carry out a thorough explainability analysis (appendix A.1) that reveals typing cues (such as typing speed, session length) and emotion self-reporting characteristics play a significant role in distinguishing different emotion self-reports. The superior performance of HACE is further backed by a theoretical explanation (appendix A.2). In summary, the key contributions of this paper are as follows,

- We describe the design and implementation of HACE, a Human-AI collaborative framework (encompassing an active learning module) to reduce the emotion self-report collection effort in long-term ESM studies. We show the implementation of the framework with a case study on smartphone keyboard interaction. We provide fine-grained details of the implementation of the active learning framework.
- To evaluate the framework, we performed a large-scale in-the-wild user study with 32 users who provided keyboard interaction data and emotion self-reports for three weeks, resulting in 203 hours of typing data. From the user study, we observed that HACE required 46% fewer self-reports as compared to the traditional self-reporting approaches. Although HACE approximately halves the number of user self-report inputs, it detects the emotion self-reports with an average F-score of 85%.
- We also performed the explainability analysis to highlight that relevant typing characteristics (e.g., speed, session length) and self-reporting patterns play an important role in distinguishing emotion self-reports.

2 RELATED WORKS

In this section, we discuss the related works in terms of the (a) emotion ground truth collection in ESM-based studies, (b) emotion self-report collection for smartphone keyboard-based emotion detection, and (c) usage of Human-AI collaborative approaches to reduce the manual labeling effort.

2.1 Emotion Ground Truth Collection in Experience Sampling Method Studies

In behavioral research, one of the most commonly adopted approaches for emotion ground truth collection is the Experience Sampling Method (ESM) [30, 36], which allows in-situ sampling of user behavior using a set of questionnaires. Traditionally, users maintained a diary entry to keep track of various events. More recently, with the proliferation of wearable devices and smartphones, the in-situ sampling of behavioral data is performed by triggering notifications [5]. In the case of emotion-related studies, researchers usually collect the emotion ground truth labels as self-reports by collecting responses from the users to these ESM probes [40, 52]. One major challenge in ESM-driven self-report collection is that the participants must respond to repetitive self-report probes over the duration of the study. As a result, the self-report collection process becomes time-consuming and labour-intensive [5, 47, 50].

Currently, to reduce the user burden, typically *time-triggered* ESM probes are scheduled at fixed intervals so that the users need to respond to fewer probes [40, 75]. However, longer time-interval can lead to missing out on key events. To avoid this, in some studies, *event-triggered* schedules are also used [19, 53]. Some researchers also recommended the usage of a hybrid schedule by combining both time-triggered and event-triggered schedule so that an ESM probe is triggered only when an event takes place and there is sufficient gap between two consecutive ESM probes [23]. In some emotion self-report collection studies, researchers have also recommended the usage of continuous emotion rating collection approach to collect more fine-grained emotion self-reports [60, 79]. However, as continuous self-report collection significantly increases user burden, researchers also recommended performing the ESM probing opportunistically, i.e., only when there is a significant variation in one of the physiological signals of the user [2].

More recently, the researchers have started to investigate different approaches of collecting emotion ground truth without collecting the self-reports. These approaches have the potential to overcome the self-reporting burden. For example, Tag et al. [65] developed smartphone application to capture the facial images and extracted the emotion captured in the images by invoking state-of-the-art facial emotion recognition tools (Affectiva API[1]). Similarly, Khalid et al. analyzed phone data to construct a social network and integrate the temporal dynamics to determine self-reported happiness and stress levels [33]. Notably, in these approaches, the emotion self-reports are inferred from alternative modalities, which may not be available in all ESM related studies.

2.2 Emotion Self-report Collection for Smartphone Keyboard Interaction Based Emotion Detection

With the proliferation of smartphones and the overwhelming usage of different instant messaging applications such as Whatsapp, and Facebook messenger, it is common for people to express their emotions frequently through these apps [37, 55, 76]. As a result, smartphone keyboard interaction patterns (not actual content) have been leveraged for unobtrusive mental health monitoring, stress measurement, and emotion-aware services [9, 26, 63, 74]. Researchers have demonstrated that keyboard interaction patterns (e.g., typing speed, touch pressure, error rate) can be used to develop machine learning models for emotion inference [22, 63, 73, 74]. At the same time, they acknowledged that collecting the emotion self-reports to develop smartphone keyboard interaction-based emotion-aware applications is challenging due to the repetitive nature of self-report collection [5, 41, 73].

To address these challenges, researchers adopted different strategies for efficient emotion self-report collection in the context of smartphone keyboard interaction. For example, Ghosh et al. developed a 2-phase ESM protocol, where the first phase generates the ESM probes based on the amount of typing performed by the user, and the second phase employs a machine learning model to issue (or skip) the ESM probe if the user attention is not available [24]. Researchers also demonstrated that by leveraging the time-domain and frequency-domain representations of typing interaction patterns, it may be possible to trigger the ESM probes at the opportune moments [27]. However, these approaches primarily concentrate to find the suitable probing moments (when user attention is available) and do not necessarily aim to reduce the number of probes that the user needs to respond. On the contrary, HACE aims to reduce the number of probes by asking the user only for the not-so-confident typing sessions.

2.3 Human-AI Collaborative Approach to Reduce the Annotation Effort

In a typical Human-AI collaborative approach, the interaction between humans and AI occurs at the following stages - (a) data producing and pre-processing, (b) ML modeling, and (c) model evaluation and refinement [43]. Among all these stages, data labeling in the first advocates for co-operations between human and AI as gathering labeled data is one of the challenging tasks while developing a machine learning model [64, 68, 69, 78].

One of the most commonly used machine learning approaches in the data labeling process is active learning, in which the key idea is that if a model is trained intelligently with informative instances, it can perform well even with less training data [57]. This idea is useful in different scenarios (e.g., image classification [8, 20], image retrieval [4], image captioning [15], interruptible moment identification [28]), where obtaining labels is expensive (time-consuming, resource-consuming) [80]. Broadly, there are two types of active learning algorithms - (a) *stream-based*, where the unlabeled samples are generated as a stream of data [10], and (b) *pool-based*, where a large pool of unlabeled samples is available [38]. In active learning, first, a base model is trained with a set of labeled samples (known as seed samples), and then the model selects 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 (e.g., query-by-committee, uncertainty sampling, mutual information based sampling) are used [13, 32, 58]. Among these, uncertainty sampling is the most widely used approach in which the model queries for the instances

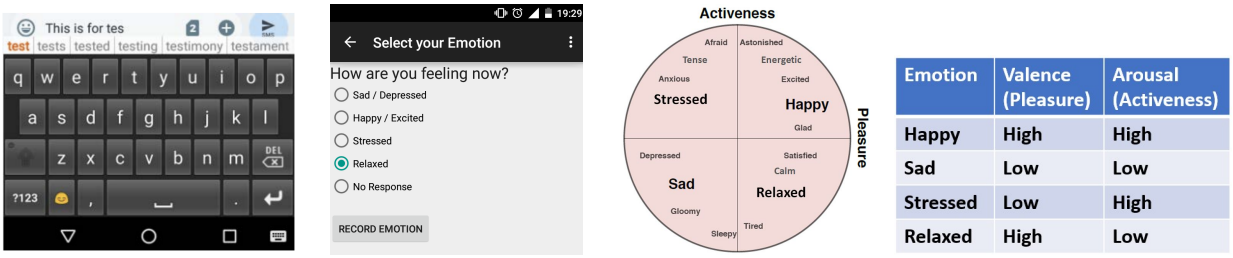
[1] https://www.affectiva.com/. Accessed: 04/24/2023.

it is most uncertain about [3, 12, 38, 58]. In the existing literature, different notions of uncertainty are used, e.g. margin [3], least confidence [12], entropy [58]. Once the label is acquired from the user, the base model is retrained (at a certain rate, based on the application) to make it more accurate for future predictions.

Key Takeaways: Summarizing the discussion on the related works, we observe that to reduce the emotion self-report collection effort in ESM-based studies, different approaches have been practiced. We also note that Human-AI collaboration approaches are used in different domains to reduce the annotation effort, although the application of such a collaborative approach is not explored in the context of emotion self-report collection. This gap in the literature introduces the opportunity for developing a Human-AI collaborative approach for efficient emotion self-report collection, which we investigate in this work.

3 CASE STUDY: SMARTPHONE KEYBOARD BASED EMOTION SELF-REPORT COLLECTION

This work focuses on efficient collection of emotion self-reports based on smartphone keyboard interaction. A keyboard is one of the most popular modalities for inputting user data into the smartphone. Several researchers have used this input modality in the past for emotion recognition [9, 22, 63]. In this section, we discuss in detail the user study in terms of experiment apparatus, data logging, and the study procedure. This work has been approved by our institute's ethics committee, and we have obtained the IRB approval prior to the user study.



Emotion	Valence (Pleasure)	Arousal (Activeness)
Happy	High	High
Sad	Low	Low
Stressed	Low	High
Relaxed	High	Low

(a) App keyboard (b) Self-reporting UI (c) Circumplex model [56] (d) Emotion and VA relationship

Fig. 2. 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 Circumplex model of emotion, which guides the self-report UI design, (d) relationship among the selected emotions and the valence-arousal (VA)

3.1 Experiment Apparatus

We developed the experiment apparatus (Fig. 2) consisting of two major components. The first component is an Android QWERTY keyboard (Fig. 2a) based on Android Input Method Editor (IME) that facilitates tracing user's keyboard interactions. This keyboard allows us to capture the user's typing pattern. One must note that to mitigate any privacy concerns, we do not store any alphanumeric character that the user inputs. The second component is an emotion self-report collection UI (Fig. 2b), which captures the emotion self-report response from the user. The self-reporting UI consists of four emotions (*happy, sad, stressed, relaxed*); the users need to select one emotion at a time based on what they are experiencing at the moment, and press the 'Record Emotion' button to log the data.

We select these emotions based on the Circumplex model (Fig. 2c) of emotion [56]. According to this model, human emotion comprises two dimensions - valence (indicating the pleasure) and arousal (indicating the activeness). As a result, the Circumplex model represents emotions in a 2D plane in four quadrants. Selecting a representative emotion from each quadrant allows to cover the different spectrum of valence and arousal. Therefore, we select these four emotions (*happy, sad, stressed, relaxed*), which belong to different quadrant of the

Circumplex plane. We show the mapping between these emotions and their valence and arousal (based on the position on the Circumplex plane) in Fig. 2d. We did not consider the neutral emotion in the self-report UI. This is primarily because neutral emotion is at the origin of the Circumplex plane, (valence = arousal = 0); therefore, maintaining the same reference for the participants in a long-term study can be challenging. Additionally, we kept the interface simple by explicitly recording the emotion. We did not consider the intensity of perceived emotion, which can make self-reporting difficult. We also keep the provision of *No Response*, so that the user can skip self-reporting by selecting this option.

3.2 Logging Keyboard Interactions and Emotion Self-reports

We next describe how we capture the user's keyboard interaction and associated emotion using the apparatus described in Section 3.1.

Tracing Keyboard Interactions: Once a user types on their phone using the app keyboard (Fig. 2a), we log details relevant to every typing session, which is defined as the time period spent by the user at-a-stretch on a single application. In specific, we capture the timestamp of every touch event in a session and then compute the elapsed time between two consecutive touch events. This interval is defined as the *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; a higher value of ITD indicates a 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 Typing Sessions: We also collect self-reported emotions from users. We probe the user at the end of a typing session (i.e., time spent on a single app at-a-stretch). In specific, once the user completes typing in an application, and switches from the current application, we probe the user for the emotion self-report. We do not probe the user even if the text input is done but the user stays on the same application (i.e., continues with the same session). The emotion self-report collector UI is shown in Fig. 2b. We associate the provided emotion self-report with the current typing session. We discard the sessions tagged with *No Response* and do not consider them in our analysis.

3.3 Study Procedure

We recruited 36 participants (24M, 12F) aged between 20 to 35 years from our university. The average age of the participants was 28.7 years (std. dev. 4.7). We installed our application on their smartphones and asked them to use it for three weeks for regular typing activities and emotion self-reporting. We also informed them that they would receive a self-report pop-up once they completed typing in an application and switched from the current application. Once the self-report pop-up was delivered to the user, it remained in the foreground. There was no timeout for the pop-up. A user could dismiss the pop-up either by recording the emotion self-report or by swiping it away. The participant was required to record their perceived emotion from one of the four available options. The participants were further instructed that if they wished to skip answering a probe, they should select the *No Response* button instead of dismissing the pop-up.

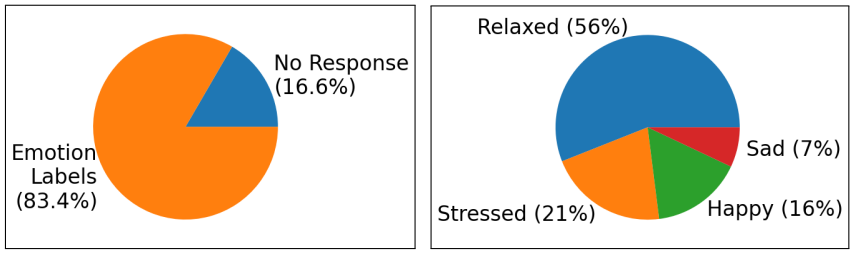
As we were interested in capturing momentary emotional variations during typing, we probed the user immediately after the typing session (i.e., the user completes typing in an application and changes the application). This allowed to capture the self-reports close to the typing sessions and reduced the possibility of emotion attenuation. To reduce this fading effect of emotion, we did not consider the self-reports, which were collected after 3 hours (the same interval was used in earlier works [22]) of a typing session. At the end of the study, we observed that out of the 36 participants, 4 participants did not provide enough self-report, i.e., they provided

less than 50 self-reported emotion states during the three weeks of usage. We did not use their data for analysis (the same number of self-report labels was used in earlier works [22]). We ran all analysis on the remaining 32 participants' data (22 M, 10 F).

4 DATA ANALYSIS: FEATURE EXTRACTION AND FEASIBILITY STUDY

4.1 Dataset Description

We collected a total of 11285 typing sessions from the participants. This corresponds to 203.5 hours of typing data. We issued on average 16.8 probes per day for every user corresponding to these typing sessions. We obtained 16.6% of *No Response* sessions, which we eliminated (See Fig. 3a). As a result, our dataset consists of 9409 typing sessions tagged with different emotion self-reports. The average number of sessions for every user is 294.03 (std. dev 124.51). The median and 75^{th} percentile session duration is 46.1 sec. and 166.3 sec. respectively. We observed most sessions (56%) were tagged with the *relaxed* emotion, whereas 21%, 16%, and 7% sessions were tagged with *stressed, happy* and *sad* emotion respectively (See Fig. 3b). The imbalance in emotion distribution can be attributed to the in-the-wild nature of the study, which does not allow inducing specific emotion. Similar findings have been reported in previous studies [22, 40] We summarize the final dataset in Table 1.



(a) *No Response* and Emotion distribution (b) Emotion-wise session distribution

Fig. 3. Distribution of typing sessions - (a) distribution of *No Response* and emotion label sessions (b) emotion-wise distribution of different typing sessions.

Table 1. Final dataset details

Total typing sessions	11285
Total typing duration (in Hr.)	203.5
No Response sessions (eliminated)	1876
Total sessions (tagged with emotions)	9409
Session duration (median, 75^{th} percentile)	46.1 sec, 166.3 sec.
User-wise avg. no of session	294.03 (SD: 124.51)
User-wise avg. no daily ESM probes	16.8 (SD: 4.17)

These findings demonstrate that a few emotions (e.g., *relaxed, stressed*) are reported more commonly. Thus, the availability of a mechanism to automatically detect the emotion for the frequently occurring sessions can avoid probing for such sessions and therefore, the total number of probes that the user responds to can be reduced. Next, we discuss the interaction characteristics that can be used to detect the emotion for a typing session.

4.2 Typing Features for Emotion Detection

We extracted the following typing features of a session S: (a) typing speed (S_{MSI}), (b) error rate (S_{Er}), (c) special character fraction (S_{Sp}), (d) session length (S_l), (e) session duration (S_d). 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 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 $S_{Er} = \frac{c}{n}$. Any non-alphanumeric character (except backspace and delete) inputted in a session is considered as a special character. If there are k number of special characters present in a session, we compute the special character fraction as $S_{Sp} = \frac{k}{n}$. The session length (S_l) is the total number of touch events in the session, and the session duration (S_d) is the difference between the last and first touch timestamp ($S_d = t_n - t_1$). To handle the inter-subject variability [66], we normalize each feature as $x' = \frac{x - min(X)}{max(X) - min(X)}$, where $X \in \{S_{MSI}, S_{Er}, S_{Sp}, S_l, S_d\}$ 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.

4.3 Self-report Transition Features for Emotion Detection

Existing literature on ESM-based emotion self-report collection suggests that often current emotion self-report is influenced by the previous emotion self-report due to the persistence effect of emotion [72]. More specifically, the emotion self-report at n^{th} session is influenced by the emotion self-report of the $(n-1)^{th}$ typing session, and this relationship can be modeled using the Discrete Time Markov Chain [62]. To capture this persistence effect, we compute the probability of different emotion self-reports for n^{th} session as shown in Fig. 4. Mathematically, we express the same as follows,

$$e_n = e_{n-1}.P \tag{1}$$

where P is the transition matrix containing the state transition probabilities and e_n denotes the probabilities of different emotions of n^{th} *session*, e_{n-1} denotes the self-report of $(n-1)^{th}$ *session*. The state space of e_i contains the set of recorded emotion states {*happy, sad, stressed, relaxed*}. To calculate the transition matrix (P), state-wise transition probabilities are calculated. To obtain the transition probability (p_{xy}) of making a transition from state x to y, the total number of transitions (n_{xy}) made from x to y should be divided by the total number of transitions (n_x) possible from x, which can be expressed as,

$$p_{xy} = \frac{n_{xy}}{n_x} \tag{2}$$

where $x, y \in$ {*happy, sad, stressed, relaxed*}. We use the four probability values (each corresponds to one of the four emotions) recorded in e_n (see Fig. 4) to estimate the emotion self-report for n^{th} session.

4.4 Feasibility Analysis: Leveraging the difference among Various Sessions

We aim to develop a machine learning model that (a) can automatically tag a typing session with the emotion based on keyboard interaction patterns, and (b) uses as few as possible emotion self-reports to train the model so that the user needs to respond to fewer self-reports (and therefore the survey fatigue is reduced).

To investigate this, we analyzed the collected dataset further to detect underlying patterns in the feature values. Since we represent every data point (session) in terms of nine features (five typing based features (Section 4.2) and four emotion self-report transition features (Section 4.3)), first we reduce the dimensionality of the data to visualize it in a 2-D plane. We apply PCA (principal component analysis) [77] on the collected dataset (by setting the number of principal components to two) and show the outcome in a scatterplot in Fig. 5. The figure reveals that for different emotions (especially *relaxed, stressed, happy*), there is an observable difference. At the same time, there is a slight overlap between *happy* and *sad* emotions. This implies that in order to discriminate among different emotion self-reports, a machine learning model may not require a large number of emotion

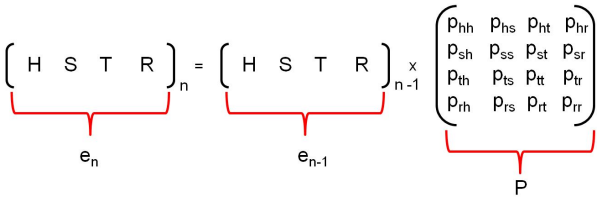


Fig. 4. Schematic showing the process of computing the probabilities of different emotions for n^{th} *session* (e_n). We multiply the self-report of $(n-1)^{th}$ *session* with transition matrix P, which is computed by analyzing the transitions of previous $(n-1)$ *sessions*. $[e_i]_{1\times4}$ is a vector with denoted position of different emotion states. For a given self-report, that position is set to 1, rest are 0. $[P]_{4\times4}$ is the transition matrix; p_{xy} indicates transition probability of moving from state x to y. *H,S,T,R* denote *happy, sad, stressed, relaxed* states respectively.

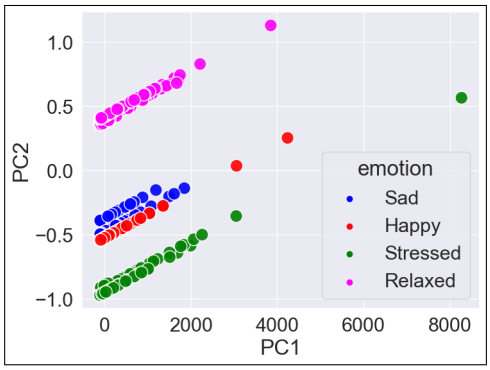


Fig. 5. The visualization reveals a noticeable difference among different typing sessions (especially *stressed, relaxed, happy*), which may be leveraged by a machine learning model to reduce the requirement of large number of emotion self-reports.

self-reports for each of the emotions. This may help to reduce the number of emotion self-reports required to train the machine learning model for emotion self-report prediction.

In summary, we observed that although we collected a large number of emotion self-reports from the participants, there are noticeable differences between typing sessions tagged with different emotions. Therefore, it may be possible to learn this difference using a machine learning model with *relatively fewer* emotion self-reports. These findings motivated us to develop the Human-AI collaborative emotion self-report collection framework as described next.

5 HACE: HUMAN-AI COLLABORATIVE EMOTION SELF-REPORT COLLECTION FRAMEWORK

In this section, we discuss the HACE framework (Fig. 6), which encompasses an active learner. To initialize the active learner, we accumulate a set of typing sessions (tagged with different emotions) as seed samples. We extract relevant features (as described in Sections 4.2, and 4.3) from these sessions and combine them with the emotion self-reports to train a machine learning model that can identify the probable emotion for an unlabelled typing session. This model is known as the base model. Now, as users perform typing activities on their phones, new typing sessions are generated. We pass every typing session through this base model to identify the probable emotion of the user during the session. If the model can confidently predict one of the four emotions (*happy, sad, stressed, relaxed*) for the session, we associate that emotion with the session and do not probe the user for the emotion self-report of that specific session. Otherwise, we probe the user for emotion self-report and retrain the base model with every new emotion self-report provided by the user. This way, we decide the emotion for every typing session and retrain the base model as often as required. This strategy not only allows reducing the number of self-reports to be answered, but also improves the learner in detecting emotion associated with every typing session by retraining. We discuss each of these steps in detail next.

Seed Sample Identification: We use a set of typing sessions, marked with emotion self-reports as seed samples. Notably, the typing sessions are generated as a stream whenever a user performs the typing activity. Therefore, we accumulate the initial x% (we set the value of x in the experimental setup, Section 6.1.2) of the typing sessions from every user and the corresponding emotion self-reports, and use those as the seed samples.

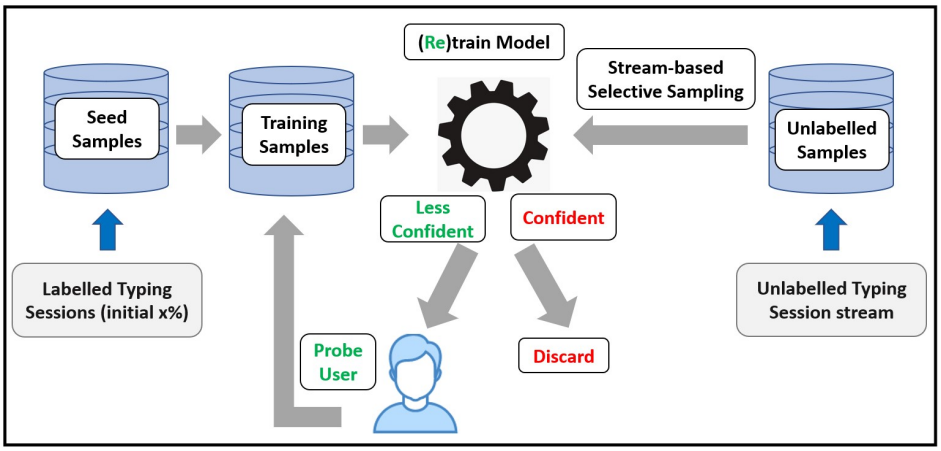


Fig. 6. The architecture of the HACE framework. First, a set of typing sessions (labeled with different emotions) are used as a seed to train the base model of the framework. After that, as new typing sessions are generated as a stream, those are sent to the model for inferring emotion. If the model is confident about the predicted emotion of the session, the user is not probed, otherwise, the user is probed for self-report. Whenever a new self-report is collected from the user, the model is retrained.

Base Model Creation: The goal of the machine learning model in the HACE framework is to automatically identify the emotion self-report for every typing session so that the number of self-report probes can be minimized. We utilize the seed samples to train a machine learning model to determine emotion self-report of a new session. From each of the typing sessions, we extract the features as mentioned in Sections 4.2, and 4.3. We train the model using a Random Forest classifier with 100 decision trees. To measure the quality of split in a tree we have used entropy for information gain. We use the default maximum depth of the tree in our implementation. This allows the nodes to expand until all leaves are pure or until all leaves contain less than the minimum number of samples (default = 2)[2]. After training the model with the seed samples, we have the base model that can be used to determine the emotion for a new unlabelled typing session.

Opportunistic Emotion Self-report Collection: Once the base model is constructed, we aim to automatically label the subsequent typing sessions using the model so that user involvement is reduced. If the model can confidently (i.e., with high probability) predict the emotion self-report for a new typing session, we avoid probing the user, thus reducing the number of probes that a user needs to respond to. However, as the typing sessions are generated as a stream, we use selective sampling [14] (as opposed to pool-based sampling) on this data stream. In specific, every newly generated typing session is sent to the base model to find its confidence to tag the current session with one of the emotion self-reports (*happy, sad, stressed, relaxed*). We adopt margin sampling [3] strategy for deciding whether to probe the user for a typing session. In this approach, the instance that has the smallest difference between the first and second most probable labels is selected for querying. More specifically, we probe the user for a typing session if the probability difference (between the first and second most probable emotion self-report) is less than a threshold, termed as margin sampling threshold (θ). We set this threshold in the experiment setup (Section 6.1.2).

Model Retraining: We retrain the base model of HACE for every user independently. In specific, for every user, the same base model is used initially. However, as a user generates the typing sessions, and the base model can

[2]https://tinyurl.com/2p9865ew. Accessed: 04/24/2023.

not confidently predict the emotion self-report for a session, the user is probed for the emotion self-report for that session. We retrain[3] the base model of HACE every time we collect an emotion self-report from the user. This allows improving the learner to detect the emotion self-reports. HACE stops the opportunistic probing and retraining once the required number of typing sessions (we set the amount of opportunistic query samples in experimental setup, Section 6.1.2) for a user are labeled. At the end of this phase, we have an improvised machine learning model for every user capable of tagging every typing session with one of the four emotion self-reports based on the typing interactions.

6 EVALUATION

In this section, we discuss the experimental evaluation of HACE. First, we describe the experiment setup, which includes the description of the baselines, evaluation strategy, and performance metrics. Later, we analyze HACE's performance in reducing the probing rate and detecting suitable probing moments. We also discuss the performance insights (in terms of sampling threshold, seed samples, and retraining) of HACE.

6.1 Experiment Setup

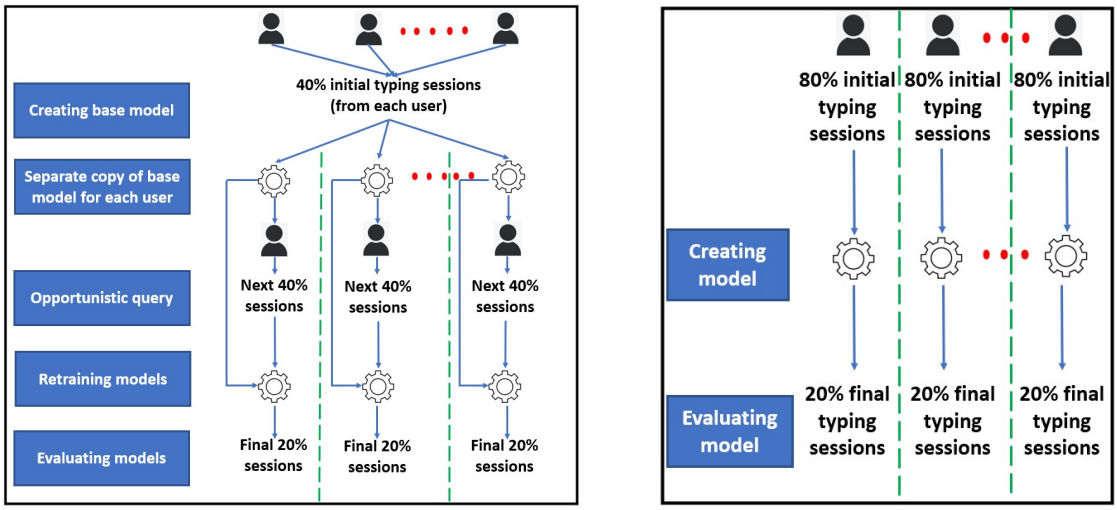
6.1.1 Baselines. The machine learning model of the HACE framework uses two sets of features – typing interaction features and self-report transition features for predicting the user's emotion during a session. So, it becomes intuitive to compare the proposed model's performance with individual sets of features. Also, we noted that the in the collected dataset, *relaxed* emotion is highly represented. Therefore, the proposed model is compared with a model that always predicts the most frequent emotion. Finally, as typing characteristics are highly personalized and earlier works demonstrated that personalized (user-dependant) models perform better than the generalized (user-independent) models [17, 22], all the baselines are personalized. In the following section, we discuss all the models that are used as baselines,

- **Typing Interaction based Model (TYP):** In this baseline, we construct a Random Forest model using *only* the typing features (typing speed, error rate, special character fraction, session length, session duration) as described in Section 4.2. This baseline is inspired by earlier works on smartphone typing based emotion detection, which demonstrates that typing interaction features can be used for emotion inference [21].
- **Self-report Transition based Model (SRT):** This baseline implements a Random Forest based machine learning model leveraging *only* the emotion self-report transition features (probability values corresponding to each emotion) as discussed in Section 4.3.
- **Combined (Comb):** This model consists of both the typing interaction and self-report transition features. It also implements a Random Forest based model.
- **Most Represented Emotion Model (MRE):** In the existing literature, for unbalanced dataset, often the model performance is compared with a model that predicts the most frequent emotion [25]. In our dataset, we observe that *relaxed* state is dominantly present. As a result, the active learner in the HACE framework needs to be compared with an emotion detection model, which always predicts the mostly represented emotion state. In this baseline, we build an emotion prediction model, which always predicts the most frequent emotion.

6.1.2 Evaluation Strategy of HACE and Baselines. In this section, we discuss the evaluation strategy of HACE and the baselines. First, we discuss the parameter value of margin sampling threshold (θ) and the different amount of data used for training, retraining, and evaluation of the models. Later, we highlight the evaluation approach with a schematic diagram (See Fig. 7).

[3]In this study, we performed the retraining offline (and not on the smartphone).

We set the value of margin sampling threshold (θ) as 0.2 for the experiments. We set the seed samples to 40% and opportunistic query samples to 40%. These values help to trade-off between probing rate reduction and emotion self-report detection performance (Section 6.4.1, 6.4.2, and 6.4.3 respectively).



(a) Evaluation approach for HACE

(b) Evaluation approach for baselines

Fig. 7. Schematic diagram for the evaluation strategy of HACE and baselines. (a) For HACE, the training, retraining, and testing of the HACE are performed in the following way. First, initial 40% seed sessions of each user are accumulated to train the base model. Second, for every user (say m), the next 40% sessions are used opportunistically to retrain the copy of the base model for that specific user. The final 20% sessions of that specific user (m) is used to evaluate the model performance for the given user. (b) For the baselines, initial 80% sessions of each user are used to train the model for that user. The final (left out) 20% typing sessions of every user are used to evaluate the model of the corresponding user.

We show the evaluation approach of HACE in Fig. 7a. To train and evaluate HACE, we split the data of every user into 3 parts - (a) seed samples, (b) opportunistic query samples, and (c) test samples. The training, retraining, and evaluation of every user are done as per the following steps. First, we combine the initial 40% typing sessions from every user and train the base model. Notably, to train the base model, data from all users are used. Second, the retraining is performed independently for every user. In specific, for every user (say $m \in M$, M implies all users), we make a copy of the base model and use the copy to determine the emotion self-reports for the next 40% typing sessions of user m. We query the user (m) only for the not-confident sessions, and retrain the copy with the newly queried self-reports. As a result, we have the final model for user m at the end of the 80% typing sessions (40% seed samples and 40% opportunistic query samples). Finally, we evaluate the model of user m using the remaining 20% typing sessions of the user m.

The evaluation approach for the baselines is shown in Fig. 7b. We trained the baselines adopting a personalized approach as typing patterns are personalized [17, 22]. In specific, we have used the initial 80% typing sessions (marked with one of the four emotions - *happy, sad, stressed, relaxed*) of each user to train the model for that user. The final (left out) 20% typing sessions of every user are used to evaluate the model of the corresponding user.

6.1.3 Performance Metrics. We use the following metrics to evaluate HACE,

Probe Reduction Rate: To compute the probe reduction rate in HACE, we determine the reduction in the number of probes responded by the users in comparison to the baselines. The baselines use self-reports

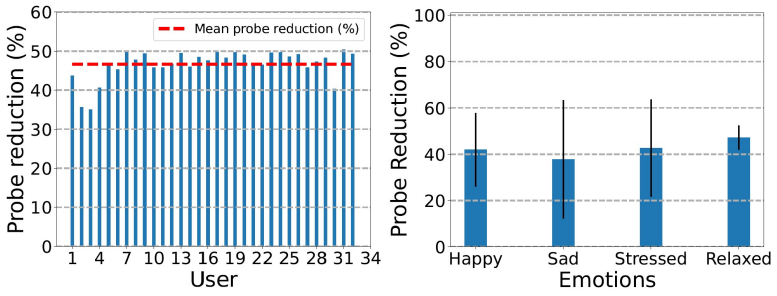
from 80% sessions for training, while HACE uses self-reports from the initial 40% (as seed) and the next 40% opportunistically. Therefore, the reduction stems from answering fewer probes from the opportunistic query samples. Specifically, in the 80% samples (40% seed, and 40% opportunistic query samples), if n_{HACE}, n_{bl} are the number of probes answered by the users in HACE and the baselines respectively, the reduction is computed as $\frac{(n_{bl}-n_{HACE})*100}{n_{bl}}$.

F-score: We use F-score as the metric to decide the emotion self-report detection performance. We compute the user-wise F-scores, which are averaged over all users to report the performance of HACE.

6.2 HACE's Performance Analysis: Probing Rate Reduction

In this section, we investigate the performance of HACE in terms of the self-report probe reduction. We present the user-wise probe reduction rate in Fig. 8a. We observe that on average there is a reduction of 46.64%. Almost 94% of the participants have a reduction of at least 40%, and almost 84% of the users have a probe reduction rate of at least 45%. This implies the effectiveness of HACE across all the users.

We also present the average probe reduction for different emotion self-reports in Fig. 8b. We observe that for each of the emotion self-reports, the average probing rate reduction is greater than 35%, while the highest reduction is observed for *relaxed* emotion (47%). Notably, in the collected dataset, we have the highest representation of the *relaxed* emotion (Section 4). Therefore, it is encouraging to observe that a large number of such self-reports can be avoided. We also observe that high amount of probe reduction for the *happy* and *stressed* emotions (≈42%) and a relatively less probe reduction for the *sad* emotion (≈38%). This can be explained by the representation of the feature values as noted in Fig. 5. We can observe that the *relaxed, stressed*, and *happy* emotion are spaced out in the feature plane, however, *sad* instances are closer to the *happy* instances. Thus, to identify these instances, the model needs more ground truths (self-reports), therefore the probe reduction for the *sad* emotions are relatively less. In summary, these findings demonstrate the effectiveness of the HACE framework in reducing the emotion self-report probes across different users and different emotions. However, the important question is that whether this probe reduction impacts the emotion self-report detection performance, which we investigate next.



(a) User-wise probe reduction (b) Emotion-wise probe reduction

Fig. 8. HACE's probing rate detection performance - (a) user-wise probe reduction (b) emotion-wise probe reduction. Error bar indicates std. dev.

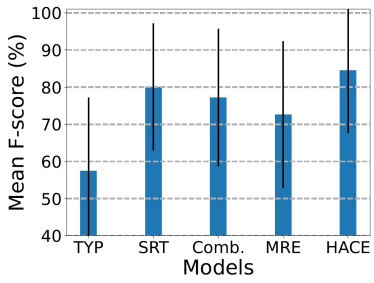


Fig. 9. HACE's emotion self-report detection performance - comparison with baseline F-score. Error bar indicates std. dev.

6.3 HACE's Performance Analysis: Emotion Self-report Detection

We next investigate the emotion self-report detection performance of the HACE framework. We compare the emotion self-report detection performance with the baselines as shown in Fig. 9. We observe that HACE outperforms all the baselines. It returns an average F-score of 84.5% (std. dev 16%). The typing only model

(**TYP**) performs the worst (average F-score: 57.5%), but the other two models (**SRT, Comb.**) perform relatively better (average F-score of 77.2%, and 81% for **Comb.** and **SRT** respectively). The poor performance of the typing only model can be attributed to the skewness of sample distribution due to which similar performance has been reported in earlier findings also [22]. However, the performance improves when the self-report transition probabilities are also considered. At the same time, we observe that although we have one emotion – *relaxed* – reported very frequently, predicting that emotion always is not a good choice, as noted in the performance of the **MRE** model (average F-score: 72.6%).

We verify the outperformance of HACE by performing a Wilcoxon test (as the user-wise F-scores do not follow normal distribution and these values are paired) for every combination. HACE performs significantly ($p<0.05$) better than **TYP** (Stat: 37.0 , p-val: 0.000), **Comb.** (Stat: 124.0 , p-val: 0.025) and **MRE** (Stat: 104.0 , p-val: 0.004) models. However, there is no significant difference for the **SRT** model (Stat: 186.0 , p-val: 0.338). In summary, HACE outperforms all the baselines in terms of the emotion self-report detection F-score even though the number of samples used for (re)training are significantly less.

6.4 HACE's Performance Insights

In this section, we analyze the performance of HACE in terms of margin sampling threshold (θ), seed samples, and the retraining opportunity.

6.4.1 Influence of Margin Sampling Threshold (θ). We assess the impact of the variation of the margin sampling threshold (θ) on emotion self-report detection performance and probe reduction in Fig. 10 for fixed amount of seed samples (40%) and opportunistic query sample (40%). We vary the value of θ from 0.1 to 0.9 and record the emotion self-report detection F-score and probe rate reduction. It is observed that when the threshold (θ) is small (i.e., the probing happens only when the difference between the probabilities of the top two predicted self-reports is very close), we end up probing fewer times, resulting in higher reduction in probing rate. This happens because in the dataset, we have relatively fewer instances, which are very close to the decision boundary (Fig. 5). But this leads to poor emotion self-report detection performance (≈81%) for θ = 0.1. However, once the value of the threshold is gradually increased, the emotion detection performance improves (although the probing reduction reduces as we start probing even if there is a large difference in the probability values of the top two predicted self-reports). However, once the value of the threshold (θ) increases beyond a limit (0.2), the emotion detection performance does not improve, but the probing reduction rate drops. Therefore, a threshold 0.2 is used in our experiments to find a trade-off between emotion self-report detection performance and probing rate reduction.

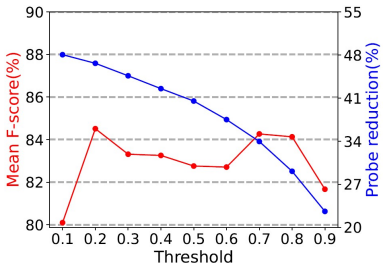


Fig. 10. Variation in emotion self-report detection performance and probe reduction with different thresholds (θ).

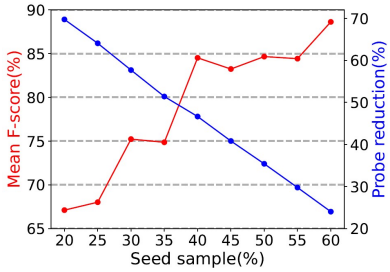


Fig. 11. Variation in emotion self-report detection performance and probe reduction with different seed samples.

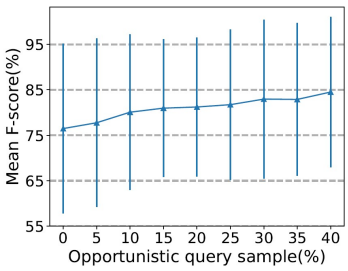


Fig. 12. Variation in emotion self-report detection performance by retraining with different opportunistic samples.

6.4.2 Influence of Seed Samples. In this section, we investigate the the amount of seed samples required to instantiate the HACE framework. To find this out, we measure the variation in emotion self-report detection performance and the probe reduction with increasing amount of seed samples in Fig. 11. We vary the amount of seed samples from 20% to 60% and record the emotion self-report detection F-score and probe rate reduction. We observe that with increasing amount of seed samples, first the emotion detection performance improves (upto 40% of seed samples) and then the performance stabilizes (beyond 40% seed samples). The F-score for self-report detection performance increases from 67% to ≈85% for 40% seed samples and beyond this, there is not much improvement in the self-report detection performance. This can be explained as follows - we need sufficient number of samples to figure out the difference among different emotion self-reports using the keyboard interaction features and the self-report transition features. However, supplying more seed samples does not help much in figuring this difference, therefore, the emotion detection performance does not improve much. On the contrary, if more seed samples are used, the probe reduction rate gradually drops (as the user needs to record more self-reports). Therefore, to trade-off between emotion self-report detection performance and probe reduction rate, around 40% instances may be used as seed samples.

6.4.3 Influence of Retraining. We also investigate the influence of retraining as it helps to improve the active learner embedded in the HACE framework. Once the basic model is ready, the query samples are used to seek human assistance opportunistically and retrain the model as required. Therefore, with opportunistic query samples, the model gets the opportunity retrain. To verify the influence of retraining, we keep the amount of seed samples fixed (40%), vary the opportunistic query samples from 0% to 40%, and evaluate the self-report detection performance on the last 20%. Notably, the amount of opportunistic query samples is not increased beyond 40%, as the amount of seed samples, and testing samples is fixed at 40% and 20% respectively. We present the variation in the emotion self-report detection performance with increasing amount of query samples in Fig. 12. When no query samples are used, there is no retraining opportunity (and the base model is used as the final model). It is observed, with no retraining opportunity, the model performs poorly with an average F-score of 76%. However, with the increasing amount of query samples, the model gets more opportunity of retraining and the emotion self-report detection performance improves. We obtain the highest mean F-score for 40% query samples.

7 DISCUSSION AND FUTURE WORKS

The empirical analysis presented earlier demonstrates that HACE framework is able to reduce the emotion self-report collection effort for long-term ESM studies. However, deploying the proposed framework for emotion self-report collection studies as well as for other ESM studies requires consideration of different practical issues, which we discuss next. We also present the limitations from the current study.

7.1 Implication of the Findings

The major implication of the findings from the study is that human-AI collaborative approaches can be applied to reduce the self-report collection effort. This area was relatively under-explored in the context of ESM based emotion self-report collection user studies. As a result, this opens up the possibilities to reduce the user burden in long-term, longitudinal user studies. This can also assist the HCI research community, who often struggles to keep the user engaged in a long-term study due to significant commitment required from the study population. Moreover, the findings also demonstrate that there is a noticeable pattern in the emotion self-report transition behavior (as observed by the performance of the **SRT** model in Section 6.3). Similar findings have also been reported in earlier literature, which suggests that the transition pattern among different emotions may be modeled by a Markov Decision Process (MDP) [67]. This can be worth investing as it may be possible to record the emotion self-reports *only* and an emotion detection model can be constructed without using any other modality (such as facial expression, speech).

7.2 Deployment Considerations

One practical aspect to consider before deploying the HACE framework is the amount of samples to be used as seed samples. Our analysis demonstrates that approximately 40% to 45% of the samples can be used as seed (Fig. 11). As a result, we envision that if an ESM-based user study is planned for 3 weeks (21 days), may be data collected from the initial 8 to 9 days can be used as seed samples, and beyond that point, the self-reports can be collected opportunistically. Another key deployment consideration is the retraining frequency. In our analysis, we have considered that whenever a self-report is collected from the user (during the opportunistic query phase), we retrain the model. However, this may not be optimal based on the training overhead (e.g., training time required, data volume). We recommend that the retraining frequency should be decided based on the overhead of the specific study. Finally, in the proposed framework, we have obtained superior performance in terms of self-report reduction rate and the self-report detection performance for a small value of the margin sampling threshold ($\theta = 0.2$). Deciding the optimal value of this threshold can be challenging. However, as discussed earlier (Section 6.4.1, Fig. 10), having a higher threshold (θ) does not yield a higher probing rate reduction. Therefore, we recommend to use a small value of this threshold ($\theta \leq 0.2$) to have a trade-off between probing rate reduction and self-report detection performance.

7.3 Generalization of HACE Framework

In this paper, we have collected emotion self-reports via ESM probes, with a specific questionnaire as shown in Fig. 2b, comprising only 4 discrete emotions (*happy, sad, stressed, relaxed*). We do not foresee significant variation in the current findings in case of other emotions, because HACE leverages the differences in typing patterns in different emotions. The existing literature demonstrates that typing patterns change with emotions [9, 17]. Therefore, as long as the user experiences different emotions (with different levels of valence and arousal), there would be variation in the keyboard interaction patterns that can be captured by the proposed framework. The selection of emotions, which are representative of different possible combinations of valence and arousal, reinforces it further that for other emotions also the proposed framework would generalize. For example, we envision that the ESM study questionnaire can be extended to more number of emotion choices (beyond just 4), multiple-choice questions, etc. Additionally, it is possible to implement other scales like Self-assessment Manikin (SAM) [6], Ekman's six basic emotion model [16] in the self-report study design. Similarly, another important question is applicability of the HACE framework in other types of ESM studies (beyond emotion self-report collection). Notably, the crux of the framework is figuring out a noticeable difference in the feature values (modalities) of different labels using as few labels as possible (as shown in Fig. 5 that noticeable difference exists among typing and self-report features for different emotions). Therefore, for any type of user study, where such differences can be figured out easily using a few labelled instances, this framework can be applied. However, the investigator responsible for the user study needs to figure out these differences and adapt the framework accordingly for the corresponding domain.

7.4 Limitations

In this section, we discuss the limitations of the HACE framework. First, in the proposed approach, the number of samples to be used as seed samples is identified empirically. In our future work, we aim to improve this by measuring the change in the distribution between observed samples and newly encountered samples with help of metrics like conditional mutual information (CMI) [31]. Second, the margin sampling threshold (θ) is empirically derived. Efficient ways to automatically derive this threshold remains to be another future work.

8 CONCLUSION

We, in this paper, propose a Human-AI Collaborative Emotion self-report collection framework, HACE, that reduces the emotion self-report collection effort in long-term ESM studies. The collaboration between human and AI happens by adopting an active learning strategy to reduce the emotion labeling effort. The active learner embedded in the framework is instantiated with a few seed samples to train a base model, which estimates the emotion self-reports for the newly generated instances adopting a selective sampling strategy. The instances are passed through the base model, and the model predicts the probable emotion self-reports. If the model is confident about the predicted outcome, we do not probe the user for emotion self-report; otherwise, we probe the user for the emotion self-report and retrain the learner to predict future emotion self-reports accurately. The number of self-report probes is reduced as we probe the user opportunistically. We evaluate HACE in the context of smartphone keyboard interaction based emotion self-report collection by running an in-the-wild study for three weeks involving 32 participants, who recorded their typing interaction details and emotions (*happy, sad, stressed, relaxed*). The empirical evaluation of the framework on the collected dataset from this study reveals that HACE reduces the average probing rate by 46% and detects the emotion self-reports with an average F-score of 85%. These findings demonstrate the possibility of using Human-AI collaborative approaches to reduce the emotion self-report collection burden in different ESM studies.

ACKNOWLEDGMENTS

The authors would like to thank BITS Goa for supporting this work under the research project 'Proposal to Develop the EdgeSys Lab at BITS Goa' sanction letter number and date GOA/ACG/2022-2023/Oct/11, Dt. 27-Oct-2022.

REFERENCES

[1] Rebecca Adaimi and Edison Thomaz. 2019. Leveraging active learning and conditional mutual information to minimize data annotation in human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 3 (2019), 1–23.

[2] Akhilesh Adithya, Snigdha Tiwari, Sougata Sen, Sandip Chakraborty, and Surjya Ghosh. 2022. OCEAN: Towards Developing an Opportunistic Continuous Emotion Annotation Framework. In *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE, 9–12.

[3] Maria-Florina Balcan, Andrei Broder, and Tong Zhang. 2007. Margin based active learning. In *International Conference on Computational Learning Theory*. Springer, 35–50.

[4] Björn Barz, Christoph Käding, and Joachim Denzler. 2018. Information-theoretic active learning for content-based image retrieval. In *German Conference on Pattern Recognition*. Springer, 650–666.

[5] Niels Van Berkel, Denzil Ferreira, and Vassilis Kostakos. 2017. The Experience Sampling Method on Mobile Devices. *ACM Computing Surveys (CSUR)* 50, 6 (2017), 93.

[6] Margaret M Bradley and Peter J Lang. 1994. Measuring emotion: the self-assessment manikin and the semantic differential. *Journal of behavior therapy and experimental psychiatry* 25, 1 (1994), 49–59.

[7] Luca Canzian and Mirco Musolesi. 2015. Trajectories of depression: unobtrusive monitoring of depressive states by means of smartphone mobility traces analysis. In *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*. 1293–1304.

[8] Jongwon Choi, Kwang Moo Yi, Jihoon Kim, Jinho Choo, Byoungjip Kim, Jinyeop Chang, Youngjune Gwon, and Hyung Jin Chang. 2021. 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*. 6749–6758.

[9] Matteo Ciman and Katarzyna Wac. 2016. Individuals' stress assessment using human-smartphone interaction analysis. *IEEE Transactions on Affective Computing* 9, 1 (2016), 51–65.

[10] David Cohn, Les Atlas, and Richard Ladner. 1994. Improving generalization with active learning. *Machine learning* 15, 2 (1994), 201–221.

[11] Sunny Consolvo and Miriam Walker. 2003. Using the experience sampling method to evaluate ubicomp applications. *IEEE Pervasive Computing* 2, 2 (2003), 24–31.

[12] Aron Culotta and Andrew McCallum. 2005. Reducing labeling effort for structured prediction tasks. In *AAAI*, Vol. 5. 746–751.

[13] Sanjoy Dasgupta and Daniel Hsu. 2008. Hierarchical sampling for active learning. In *Proceedings of the 25th international conference on Machine learning*. 208–215.

[14] Ofer Dekel, Claudio Gentile, and Karthik Sridharan. 2012. Selective sampling and active learning from single and multiple teachers. *The Journal of Machine Learning Research* 13, 1 (2012), 2655–2697.

[15] Yue Deng, KaWai Chen, Yilin Shen, and Hongxia Jin. 2018. Adversarial Active Learning for Sequences Labeling and Generation.. In *IJCAI*. 4012–4018.
[16] Paul Ekman. 1992. An argument for basic emotions. *Cognition & emotion* 6, 3-4 (1992), 169–200.
[17] Clayton Epp, Michael Lippold, and Regan L Mandryk. 2011. Identifying emotional states using keystroke dynamics. In *Proceedings of the sigchi conference on human factors in computing systems*. 715–724.
[18] Joel E Fischer, Chris Greenhalgh, and Steve Benford. 2011. Investigating episodes of mobile phone activity as indicators of opportune moments to deliver notifications. In *Proceedings of ACM MobileHCI*. 181–190.
[19] Jon Froehlich, Mike Y Chen, Sunny Consolvo, Beverly Harrison, and James A Landay. 2007. MyExperience: a system for in situ tracing and capturing of user feedback on mobile phones. In *Proceedings of the ACM Mobisys*.
[20] Weijie Fu, Meng Wang, Shijie Hao, and Xindong Wu. 2018. Scalable active learning by approximated error reduction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1396–1405.
[21] Surjya Ghosh, Niloy Ganguly, Bivas Mitra, and Pradipta De. 2017. Evaluating effectiveness of smartphone typing as an indicator of user emotion. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 146–151.
[22] Surjya Ghosh, Niloy Ganguly, Bivas Mitra, and Pradipta De. 2017. 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*. 1–12.
[23] Surjya Ghosh, Niloy Ganguly, Bivas Mitra, and Pradipta De. 2017. Towards Designing an Intelligent Experience Sampling Method for Emotion Detection. In *Proceedings of the IEEE CCNC*.
[24] Surjya Ghosh, Niloy Ganguly, Bivas Mitra, and Pradipta De. 2019. Designing an experience sampling method for smartphone based emotion detection. *IEEE Transactions on Affective Computing* (2019).
[25] Surjya Ghosh, Shivam Goenka, Niloy Ganguly, Bivas Mitra, and Pradipta De. 2019. Representation learning for emotion recognition from smartphone keyboard interactions. In *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 704–710.
[26] Surjya Ghosh, Kaustubh Hiware, Niloy Ganguly, Bivas Mitra, and Pradipta De. 2019. Does emotion influence the use of auto-suggest during smartphone typing?. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*. 144–149.
[27] Surjya Ghosh, Salma Mandi, Bivas Mitra, and Pradipta De. 2021. Exploring Smartphone Keyboard Interactions for Experience Sampling Method driven Probe Generation. In *26th International Conference on Intelligent User Interfaces*. 133–138.
[28] Surjya Ghosh, Bivas Mitra, et al. 2022. ALOE: Active Learning based Opportunistic Experience Sampling for Smartphone Keyboard driven Emotion Self-report Collection. In *2022 10th International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 1–8.
[29] Muhammad Asif Hasan, Nurul Fazmidar Mohd Noor, Siti Soraya Binti Abdul Rahman, and Mohammad Mustaneer Rahman. 2020. The transition from intelligent to affective tutoring system: a review and open issues. *IEEE Access* 8 (2020), 204612–204638.
[30] Joel M Hektner, Jennifer A Schmidt, and Mihaly Csikszentmihalyi. 2007. *Experience sampling method: Measuring the quality of everyday life*. Sage.
[31] Alex Holub, Pietro Perona, and Michael C Burl. 2008. Entropy-based active learning for object recognition. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 1–8.
[32] Sheng-Jun Huang, Rong Jin, and Zhi-Hua Zhou. 2014. Active learning by querying informative and representative examples. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 10 (2014), 1936–1949.
[33] Maryam Khalid and Akane Sano. 2023. Exploiting social graph networks for emotion prediction. *Scientific Reports* 13, 1 (2023), 6069.
[34] Ruhul Amin Khalil, Edward Jones, Mohammad Inayatullah Babar, Tariqullah Jan, Mohammad Haseeb Zafar, and Thamer Alhussain. 2019. Speech emotion recognition using deep learning techniques: A review. *IEEE Access* 7 (2019), 117327–117345.
[35] Auk Kim, Woohyeok Choi, Jungmi Park, Kyeyoon Kim, and Uichin Lee. 2019. Predicting opportune moments for in-vehicle proactive speech services. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*. 101–104.
[36] Reed Larson and Mihaly Csikszentmihalyi. 1983. The experience sampling method. *New Directions for Methodology of Social & Behavioral Science* (1983).
[37] Uichin Lee, Joonwon Lee, Minsam Ko, Changhun Lee, Yuhwan Kim, Subin Yang, Koji Yatani, Gahgene Gweon, Kyong-Mee Chung, and Junehwa Song. 2014. Hooked on smartphones: an exploratory study on smartphone overuse among college students. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 2327–2336.
[38] David D Lewis and William A Gale. 1994. A sequential algorithm for training text classifiers. In *SIGIR'94*. Springer, 3–12.
[39] Shan Li and Weihong Deng. 2020. Deep facial expression recognition: A survey. *IEEE transactions on affective computing* (2020).
[40] Robert LiKamWa, Yunxin Liu, Nicholas D Lane, and Lin Zhong. 2013. MoodScope: building a mood sensor from smartphone usage patterns. In *Proceeding of the ACM Mobisys*. 389–402.
[41] Hong Lu, Denise Frauendorfer, Mashfiqui Rabbi, Marianne Schmid Mast, Gokul T Chittaranjan, Andrew T Campbell, Daniel Gatica-Perez, and Tanzeem Choudhury. 2012. Stresssense: Detecting stress in unconstrained acoustic environments using smartphones. In *Proceedings*

of the 2012 ACM conference on ubiquitous computing. 351–360.
[42] Scott M Lundberg and Su-In Lee. 2017. 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.). 4765–4774.
[43] Mansoureh Maadi, Hadi Akbarzadeh Khorshidi, and Uwe Aickelin. 2021. A review on human–AI interaction in machine learning and insights for medical applications. *International journal of environmental research and public health* 18, 4 (2021), 2121.
[44] Abhinav Mehrotra, Sandrine R Müller, Gabriella M Harari, Samuel D Gosling, Cecilia Mascolo, Mirco Musolesi, and Peter J Rentfrow. 2017. Understanding the role of places and activities on mobile phone interaction and usage patterns. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 1–22.
[45] 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*.
[46] 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*. 1021–1032.
[47] Abhinav Mehrotra, Jo Vermeulen, Veljko Pejovic, and Mirco Musolesi. 2015. Ask, but don't interrupt: the case for interruptibility-aware mobile experience sampling. In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*. ACM, 723–732.
[48] Prasanth Murali, Javier Hernandez, Daniel McDuff, Kael Rowan, Jina Suh, and Mary Czerwinski. 2021. Affectivespotlight: Facilitating the communication of affective responses from audience members during online presentations. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–13.
[49] Veljko Pejovic, Neal Lathia, Cecilia Mascolo, and Mirco Musolesi. 2016. Mobile-based experience sampling for behaviour research. In *Emotions and Personality in Personalized Services*. Springer, 141–161.
[50] 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.
[51] 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 ACM SIGCHI*. 3319–3328.
[52] 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 ACM UbiComp*. 825–836.
[53] Kiran K. Rachuri, Mirco Musolesi, Cecilia Mascolo, Peter J. Rentfrow, Chris Longworth, and Andrius Aucinas. 2010. EmotionSense: A Mobile Phones Based Adaptive Platform for Experimental Social Psychology Research. In *Proceedings of ACM UbiComp*.
[54] Andrew Raij, Animikh Ghosh, Santosh Kumar, and Mani Srivastava. 2011. 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.
[55] Mintra Ruensuk, Eunyong Cheon, Hwajung Hong, and Ian Oakley. 2020. How do you feel online: Exploiting smartphone sensors to detect transitory emotions during social media use. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 4 (2020), 1–32.
[56] James A Russell. 1980. A circumplex model of affect. *Journal of Personality and Social Psychology* 39, 6 (1980), 1161–1178.
[57] Burr Settles. 2009. Active learning literature survey. *University of Wisconsin-Madison Department of Computer Sciences* (2009).
[58] Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. 1070–1079.
[59] Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press. I–XVI, 1–397 pages.
[60] Karan Sharma, Claudio Castellini, Egon L van den Broek, Alin Albu-Schaeffer, and Friedhelm Schwenker. 2019. A dataset of continuous affect annotations and physiological signals for emotion analysis. *Scientific data* 6, 1 (2019), 1–13.
[61] Lin Shu, Jinyan Xie, Mingyue Yang, Ziyi Li, Zhenqi Li, Dan Liao, Xiangmin Xu, and Xinyi Yang. 2018. A review of emotion recognition using physiological signals. *Sensors* 18, 7 (2018), 2074.
[62] William J. Stewart. 1994. *Introduction to the numerical solution of Markov chains*. Princeton Univ. Press, Princeton, NJ. http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+152880593&sourceid=fbw_bibsonomy
[63] Yoshihiko Suhara, Yinzhan Xu, and Alex'Sandy' Pentland. 2017. Deepmood: Forecasting depressed mood based on self-reported histories via recurrent neural networks. In *Proceedings of the 26th International Conference on World Wide Web*. 715–724.
[64] Boyuan Sun, Qiang Ma, Shanfeng Zhang, Kebin Liu, and Yunhao Liu. 2015. iSelf: Towards cold-start emotion labeling using transfer learning with smartphones. In *IEEE INFOCOM*. 1203–1211.
[65] Benjamin Tag, Zhanna Sarsenbayeva, Anna L Cox, Greg Wadley, Jorge Goncalves, and Vassilis Kostakos. 2022. Emotion trajectories in smartphone use: Towards recognizing emotion regulation in-the-wild. *International Journal of Human-Computer Studies* 166 (2022).
[66] Ronnie Taib, Jeremy Tederry, and Benjamin Itzstein. 2014. Quantifying driver frustration to improve road safety. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. 1777–1782.
[67] Mark A Thornton and Diana I Tamir. 2017. Mental models accurately predict emotion transitions. *Proceedings of the National Academy of Sciences* 114, 23 (2017), 5982–5987.

[68] Takamichi Toda, Sozo Inoue, Shota Tanaka, and Naonori Ueda. 2014. Training Human Activity Recognition for Labels with Inaccurate Time Stamps *(UbiComp '14 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 863–872. https://doi.org/10.1145/2638728.2641297

[69] Dorra Trabelsi, Samer Mohammed, Faicel Chamroukhi, Latifa Oukhellou, and Yacine Amirat. 2013. An Unsupervised Approach for Automatic Activity Recognition Based on Hidden Markov Model Regression. *IEEE Trans Autom. Sci. Eng.* 10, 3 (2013), 829–835. http://dblp.uni-trier.de/db/journals/tase/tase10.html#TrabelsiMCOA13

[70] Liam D Turner, Stuart M Allen, and Roger M Whitaker. 2015. Interruptibility prediction for ubiquitous systems: conventions and new directions from a growing field. In *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*.

[71] Liam D Turner, Stuart M Allen, and Roger M Whitaker. 2015. Push or delay? decomposing smartphone notification response behaviour. In *Human Behavior Understanding*. Springer, 69–83.

[72] Philippe Verduyn and Saskia Lavrijsen. 2015. Which emotions last longest and why: The role of event importance and rumination. *Motivation and Emotion* 39, 1 (2015), 119–127.

[73] Katarzyna Wac, Matteo Ciman, and Ombretta Gaggi. 2015. iSenseStress: Assessing Stress Through Human-Smartphone Interaction Analysis. In *9th International Conference on Pervasive Computing Technologies for Healthcare-PervasiveHealth*. 84–91.

[74] Rafael Wampfler, Severin Klingler, Barbara Solenthaler, Victor R Schinazi, and Markus Gross. 2020. 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*. 1–13.

[75] Rui Wang, Fanglin Chen, Zhenyu Chen, Tianxing Li, Gabriella Harari, Stefanie Tignor, Xia Zhou, Dror Ben-Zeev, and Andrew T Campbell. 2014. StudentLife: assessing mental health, academic performance and behavioral trends of college students using smartphones. In *Proceedings of the ACM UbiComp*. 3–14.

[76] Sophie F Waterloo, Susanne E Baumgartner, Jochen Peter, and Patti M Valkenburg. 2018. Norms of online expressions of emotion: Comparing Facebook, Twitter, Instagram, and WhatsApp. *new media & society* 20, 5 (2018), 1813–1831.

[77] Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems* 2, 1-3 (1987), 37–52.

[78] Omar F. Zaidan and Chris Callison-Burch. 2011. Crowdsourcing Translation: Professional Quality from Non-Professionals. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, 1220–1229. https://aclanthology.org/P11-1122

[79] Tianyi Zhang, Abdallah El Ali, Chen Wang, Alan Hanjalic, and Pablo Cesar. 2020. Rcea: Real-time, continuous emotion annotation for collecting precise mobile video ground truth labels. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*.

[80] Xiaojin Jerry Zhu. 2005. Semi-supervised learning literature survey. *Technical Report, University of Wisconsin-Madison Department of Computer Sciences* (2005).

[81] Manuela Züger and Thomas Fritz. 2015. Interruptibility of software developers and its prediction using psycho-physiological sensors. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 2981–2990.

A APPENDICES

In this section, we perform the explainability analysis of the HACE framework. The objective of the explainability analysis is two-fold. First, we investigate empirically to find out the significant features that are leveraged to develop the HACE framework (Appendix A.1). Later, we provide a theoretical interpretation for the superior performance of HACE (Appendix A.2).

A.1 Empirical Approach for Explainability Analysis

We performed the explainability analysis of the HACE framework using (SHapley Additive exPlanations) [42]. The shapley index of a feature is considered as its importance in deciding the outcome for an instance. We compute the shapley index for each of the features and present their role in predicting each of the emotion self-reports in Fig. 13. It is observed that among the self-report transition features, the $prob_{relaxed}$, and $prob_{stressed}$ are having the highest discriminating power, whereas among the typing features session length and MSI are the most discriminating ones. Similar findings have been reported in earlier works [21, 22] that highlight features like typing speed (a representation of MSI), session length, special character usage, persistent emotion (a representation of emotion transition characteristics) vary across different emotion and therefore can be leveraged to train an human-AI collaborative model using relatively fewer emotion self-reports. The validation from domain knowledge thus confirms the consistency of feature explainability and the prediction model towards optimal performance.

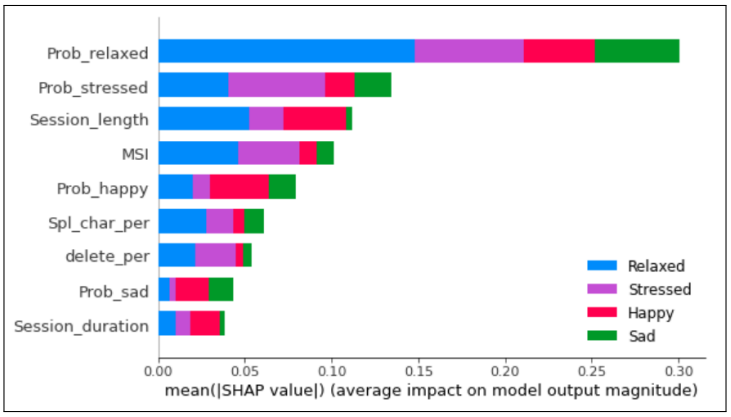


Fig. 13. Explainability analysis using SHAP reveals that among the self-report transition features, the $prob_{relaxed}$, and $prob_{stressed}$ are having the highest discriminating power, whereas among the typing features session length and MSI are the most discriminating ones

Since an active learning model is followed to ensure robust prediction, an explanation of greater prediction power (active learning model) is in order. Let us consider a scenario where the given are two different learning paradigms, same data set and different outcomes of the learning paradigms where the dataset consists of 11000 training samples with an average of 300 samples per user and 32 total number of users considered in the experiment. Each instance is a questionnaire response with 9 features. The paradigm 1 is a standard learning model where the train-test split is 4:1. We propose an active learning paradigm, paradigm 2 where the training, active learning and test splits are in 2:2:1 ratio. We explain why learning paradigm 2 outperforms learning paradigm 1 with identical sample complexity. Using the compact notation, we present a detailed mathematical argument and theoretical interpretation of the superior prediction power of the model in Appendix A.2.

A.2 Theoretical Interpretation for Explainability Analysis

We provide a theoretical explanation for the superior performance of the HACE framework. We have the four emotions $\mathbb{E} = \{h, s, t, r\} = \{0, 1, 2, 3\}$, where *happy, sad, stressed,* and *relaxed* emotions are denoted by *h, s, t (0,1.2)*, and *r (3)* respectively. The domain of the active learner in the HACE framework comprises of feature set: $\mathbb{F} = \{w_1, w_2, ..w_9\}$, nine features (as denoted in Sections 4.2, and 4.3), emotion self-reports: $\mathbb{E} = \{h, s, t, r\}$, training samples: $\mathbb{S} = \{(x_1^i, y_1), (x_2^i, y_2), ..(x_m^i, y_m)\}$, where $x_j^i \in \mathbb{F}$, $y_j \in \mathbb{E}$. Notably, $\frac{2}{5} * n \leq |S| = m \leq \frac{4}{5} * n$, where n is the total number of samples. Next, we fix some mathematical notations and definitions to describe the learning framework.

A class of predictors is PAC (Probably Approximately Correct) learnable if $\exists$ a function of sample complexity $m_H : (0, 1)x(0, 1) \rightarrow N$ such that a learner A inputs a sample S and outputs a hypothesis $h \in H$ i.e. $A : U_m(Xx(0, 1, 2, 3))^m \rightarrow f : f : X \rightarrow (0, 1, 2, 3)$ representing the union of all sample sizes that covers the sequence of examples with labels. Moreover, for every unknown probability Distribution D, labelling function $f \in H$ and $S \subset (X)$, the probability of true loss $L(D.f)$ being greater than a success threshold ϵ is bounded by the model confidence, δ. The hypothesis function for the active learning model h(s):$\{w_1, w_2, ..w_9\} \rightarrow \{h, s, t, r\}$, h(s) performs better, in the PAC sense, than other hypotheses with same sample complexities and features. $L[D, f(h(s))] < \min_{h \in H} L[D, f(h)]$, where $h(X) = h_S(X)$. The hypothesis for the baseline model h_b:$\{w_1, w_2, ..w_9\} \rightarrow \{h, s, t, r\}$. What we imply is that the Active Learner (AL) learns with tighter confidence bound.

Goal: The goal of the active learning model is to find a hypothesis h(s) belonging to H (the hypothesis class) which has a) Low generalization error $L[D, f] + L_S$, (b) minimized number of label queries, and c) high model confidence in the predicted outcomes.

Assumptions: (a) the human annotator does not mislabel, (b) let the confidence threshold be θ. If the confidence for the predicted outcome for an unlabelled instance is lesser than θ, query the user, (c) the samples are independently and identically distributed and sampled. We need to prove the superior performance of the proposed active learning model.

Argument: The active learning model h(S) works in the following way. For each user input, the first $\frac{2}{5}$th is used for finding h(s). If the next $\frac{2}{5}$th part is having a label confidence $C_i < \theta$, the training sample S is updated as $S = S \cup (x_i, y_i)$, where θ is a pre-set label confidence threshold, and the new hypothesis h(s*) is calculated. So, $L[S(h(s*))] < L_s[(h(s)]$ If $C_i \geq \theta$, S is not updated.

When the test data (i.e.), the last $\frac{1}{5}$th part is used to evaluate the model, it is obvious that the prediction confidence has increased, decreasing the true loss L[D,f] of the updated model h(s *) i.e. $L[D, f(hs*)] \leq L[D, f(h(s))]$. This is because the updated h(s*) hypothesis is retrained with an updated S to mitigate bad samples hit by previous hypothesis h(s). For any $x \in X$ for which the label confidence is lesser than a threshold θ, we are getting the actual ground truth and retraining. It makes the model more robust, and capable of patching its bad example induced misclassification.

In every iteration, since the empirical loss L[s (h(s*))] and true loss L[D,f(h(s*))] is improving in tandem, the model is not likely to overfit. Since the model uses almost minimum number of sample points to train the model, we can approximate the sample size m as $m \approx m_H$ (Sample complexity). Uncertain instances are the most informative to the model. Even if the active learner model uses the same sample size as the passive learner models, it is still choosing the training instances in priority to closely represent the unknown probability distribution D. This is not the case with the passive learner baseline models.

The baseline hypothesis h_b is trained on $\frac{4}{5}$th samples directly. Here, $ERM_H(S) = L_S[(hb)]$ $\arg\min_{h \in H}[L_s] \in (h) L[S(h_b)]$ could even be 0 i.e., $L_S[(h_b)] = 0$. But there is a possibility that $L[D, f(h_b)] > \epsilon$, resulting in overfitting. This is not the case with the active learner model. In other words, δ, the model confidence is tighter in the active learning making the success threshold, epsilon smaller in comparison to the baseline models. That is, for two competing models (H1: baseline, H2: proposed AL model), assume we have equal sample complexity $m_H(\delta, \epsilon)$, i.e. $m_{H1} = m_{H2} = m$ (say). By PAC (Probably Approximately Correct), learnability of both H1 and H2, we have the probability that the model's success is determined by (with a confidence of at least $1 - \delta$), for $\delta \in (0, \frac{1}{2})$; $P(L(D, f) > \epsilon) \leq |H| 2e^{-2m\epsilon^2} < \delta$. But δ for H1 and H2 are different, i.e., $\delta_{H1} \geq \delta_{H2} \implies 1 - \delta_{H1} \leq 1 - \delta_{H2}$. Therefore, $\delta_{H1} > |H1| 2e^{-2m\epsilon_1^2} > \delta_{H2} > |H2| 2e^{-2m\epsilon_2^2}$. Assuming $|H1| \approx |H2|$, i.e., the cardinality of the hypothesis spaces $H1$ and $H2$ are approximately same (i.e. the VC dimension of $H1$ and $H2$ are approximately same). it is easy to follow that $\epsilon_1 > \epsilon_2$, i.e., error (or the risk of misclassification) in the AL model $H2$ is smaller.

The above argument can be be extended to agnostic PAC learning [59] i.e. the AL may learn in an agnostic manner (the distribution D is defined over $X * Y$ where $Y = E$) towards the distribution of the data-labels. This, in turn, means that AL helps learn the best labeling function f by making no realizability assumptions about the label distribution but imparts additional confidence to the model by annotating correct labels on a subset of samples.