



Emotion detection from touch interactions during text entry on smartphones

Surjya Ghosh^{*,a}, Kaustubh Hiware^a, Niloy Ganguly^a, Bivas Mitra^a, Pradipta De^b

^a Department of Computer Science and Engineering, IIT Kharagpur, India

^b Department of Computer Sciences, Georgia Southern University, USA



ARTICLE INFO

Keywords:

Emotion detection

Typing

Swyping

Smartphone

Touch interaction

ABSTRACT

There are different modes of interaction with a software keyboard on a smartphone, such as typing and swyping. Patterns of such touch interactions on a keyboard may reflect emotions of a user. Since users may switch between different touch modalities while using a keyboard, therefore, automatic detection of emotion from touch patterns must consider both modalities in combination to detect the pattern. In this paper, we focus on identifying different features of touch interactions with a smartphone keyboard that lead to a personalized model for inferring user emotion. Since distinguishing typing and swyping activity is important to record the correct features, we designed a technique to correctly identify the modality. The ground truth labels for user emotion are collected directly from the user by periodically collecting self-reports. We jointly model typing and swyping features and correlate them with user provided self-reports to build a personalized machine learning model, which detects four emotion states (*happy, sad, stressed, relaxed*). We combine these design choices into an Android application *TouchSense* and evaluate the same in a 3-week in-the-wild study involving 22 participants. Our key evaluation results and post-study participant assessment demonstrate that it is possible to predict these emotion states with an average accuracy (AUCROC) of 73% (std dev. 6%, maximum 87%) combining these two touch interactions only.

1. Introduction

In smartphones, touch interactions using software keyboards is still one of the most common interfaces to interact with an application. There exist several smartphone applications, like Instant Messaging, where users interact predominantly using soft keyboards. Such keyboard touch activities can reveal important clues about the affective state of the user. It has been shown that studying keyboard activity of users on desktop computers can be effective in emotion detection (Epp et al., 2011; Kolakowska, 2013). Therefore, it is imperative that smartphone touch behavior, which is used more extensively than on desktop computers, would be as effective in emotion detection. However, unlike standard keyboards, users interacting with soft keyboard switch between different touch interactions, i.e. *typing* and *swyping* during text entry (Jiang et al., 2015). Therefore, it is important to consider both typing and swyping activities in order to infer emotion states of users from soft keyboard interactions.

There have been prior attempts on emotion recognition based on users' touch interactions on smartphones. Touch activity can be of various types - finger-stroke, tapping, drag-and-drop which are often used in gameplay, while typing and swyping commonly used for text

entry in soft keyboards. Lee et al. designed a Twitter client app and collected data from typing activity to predict emotion (Lee et al., 2012). Gao et al. analyzed finger-stroke features during gameplay on an iPhone to infer users' affect (Gao et al., 2012). Ciman et al. detected stress conditions by analyzing multiple features from swype, scroll and text input interactions (Wac et al., 2015). Ghosh et al. focused on only typing behavior to classify multiple emotion states (Ghosh et al., 2017a). In addition, recent endeavours focused on collecting contextual information and relied on smartphone sensors to detect emotion. For instance, Kim et al. proposed an emotion recognition framework from touch engagements using 12 attributes from 3 on-board sensors (accelerometer, gyroscope, touch panel) (Kim and Choi, 2012). Trojahn et al. verified multiple hypotheses on positive and negative emotion using typing speed, error rate, pressure, and context variables like gender, age, education level etc (Trojahn et al., 2013). In this paper, we contribute to the state-of-the-art by highlighting the combined influence of typing and swyping interactions for multi-state emotion detection irrespective of any other contextual information.

However, there exist challenges in considering multiple touch based input modalities, viz. typing, and swyping, during soft keyboard use. First, a user may switch between different modalities during a single session of

* Corresponding author.

E-mail addresses: surjya.ghosh@iitkgp.ac.in (S. Ghosh), hiwarekaustubh@iitkgp.ac.in (K. Hiware), niloy@cse.iitkgp.ernet.in (N. Ganguly), bivas@cse.iitkgp.ernet.in (B. Mitra), pde@georgiasouthern.edu (P. De).

<https://doi.org/10.1016/j.ijhcs.2019.04.005>

Received 29 August 2018; Received in revised form 26 March 2019; Accepted 25 April 2019

Available online 27 April 2019

1071-5819/ © 2019 Elsevier Ltd. All rights reserved.

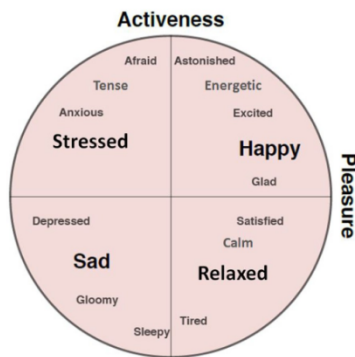


Fig. 1. Circumplex emotion model.

keyboard interaction. Identifying the correct input mode based on indirect measurement is important since that dictates the selection of the relevant characteristics of the user action. The problem compounds as there exists no pre-determined usage pattern of typing and swiping during text entry, and both of these interactions can be interspersed in any order in a text input session. Second, the usage preferences and the volume of typing and swiping vary across users. As a result, a generic method to trace individual user preference may not be effective.

In this paper, we design, implement and deploy an Android application *TouchSense*, which detects multiple emotion states based on only touch interactions performed during text entry. We determine four emotion states - *happy*, *sad*, *stressed* and *relaxed*, chosen from different quadrants of the Circumplex Model (Fig. 1) (Russell, 1980). The proposed solution traces the touch interactions by instrumenting the smartphone keyboard and distinguishes between typing and swiping interactions using a clustering method. However, we collect only the meta information, without any text content, to avoid privacy concerns. We extract features like typing speed, touch pressure, error rate from touch interactions and correlate them with self-reported emotion labels to build the personalized machine learning model for emotion detection. Self-report collection is driven by an Experience Sampling Method (ESM) (Consolvo and Walker, 2003; Hektner et al., 2007), widely used for collecting self-reports in different human studies.

We conducted the field study for 3 weeks in-the-wild by installing *TouchSense* in smartphones of 30 volunteers, of which 22 provided adequate data for analysis. *TouchSense* transparently recorded the touch interaction metadata as the users performed different text entry activities on their phone, while at the same time collected emotion self-reports from the users. We developed a personalized machine learning model for every user to classify the four emotion states and obtained an average accuracy (AUCROC) of 73%. Moreover, each emotion state is detected with an average AUCROC of at least 70%.

In summary, the main contributions of this paper are:

- We proposed and implemented a personalized machine learning model that can infer the emotion states of a user based on touch interactions during text entry.
- We conducted an uncontrolled study that collected data from a large set of users using our application, called *TouchSense*. We used the data to show the efficacy of the personalized model in predicting multiple emotion states of any user.

The rest of the paper is organized as follows. We discuss the related literature in Section 2. We outline the design and implementation of the touch-based emotion detection application *TouchSense* in Section 3, 4 respectively. The deployment of *TouchSense* and the field study are discussed in 5. We evaluate the proposed solution and perform a post-study user survey in Section 6. Finally, we discuss the implications and deployment recommendation in Section 7 before concluding in Section 8.

2. Related work

Multiple studies have aimed to determine emotion states based on smartphone activities (like SMS, call pattern, voice data), additional sensor readings (like wrist sensor, skin conductor) and contextual details (like location, proximity) (Bogomolov et al., 2014; 2013; Lam et al., 2015; LiKamWa et al., 2013; Lu et al., 2012; Pielot et al., 2015; Rachuri et al., 2010; Roshanaei et al., 2017; Sano and Picard, 2013; Politou et al., 2017). This has led to the development of different interactive recommender application based on the emotion responses (Andjelkovic et al., 2018; McGaugh, 2016). Broadly, these emotion detection explorations can be divided into two groups - (a) studies, which aim to determine a specific emotion like happiness, stress, boredom (Bogomolov et al., 2014; 2013; Lu et al., 2012; Pielot et al., 2015; Sano and Picard, 2013) and (b) studies, which determine multiple emotion states or daily average mood in terms of valence and arousal (LiKamWa et al., 2013; Rachuri et al., 2010; Roshanaei et al., 2017). Although these studies establish the utility of smartphone for emotion detection, they rely on privacy sensitive information like call patterns, use intrusive additional setup or monitors resource-expensive modality like GPS or audio.

Widespread availability of touch-based devices and steady increase (Lee et al., 2014) in the usage of instant messaging apps open a new possibility of inferring emotion from touch interactions in smartphone. In Table 1, we compare different touch-based emotion detection explorations using smartphone in terms of (a) type of touch interaction (b) usage of additional sensors (c) dependency on specific application and (d) number of detected emotion states. Gao et al. (2012); Tikadar et al. (2017) explored touch interactions during gameplay to detect emotion states. Lee et al. (2012) used sensor details and typing details in a specific application (Twitter) to determine emotion. Ciman and Wac (2016); Wac et al. (2015) explored different touch gestures like tap, scroll, swipe and text input for search and write tasks to determine stress. Kim and Choi (2012); Mottelson and Hornbæk (2016) use multiple sensor details along with touch data to determine the emotion states. In our earlier work, we focussed only on typing interactions to detect multiple emotion states (Ghosh et al., 2017a). All these works emphasize the importance of touch interactions in smartphone for emotion detection, but they (a) focus on specific application or (b) aim to determine a specific emotion state or (c) use additional modalities along with touch interactions for emotion detection.

On the contrary, our proposed approach used in *TouchSense* makes several advancements in touch-based emotion detection techniques based on smartphone. First, it does not concentrate on any specific application and focus on touch interactions during text entry in general. Second, it investigates the influence of these touch interactions for multiple emotion state detection. Finally, it does not use any other privacy sensitive additional information sources like call log, SMS details, browsing history or additional sensor details so that the proposed method can be realized to a scalable application.

3. Touchsense design

The design principles of *TouchSense* are driven by the touch based emotion detection scenario as described in Fig. 2. We define a text entry session as the time period one stays onto a single application without changing the same. In Fig. 2, elapsed time between t_1 to t_3 is defined as a session. The text entry starts with typing (t_1 to t_2) and ends with swiping (t_2 to t_3). Notably, there may be multiple typing and swiping events within a session, interleaved in all possible way; some of the representative cases are shown in Fig. 3a and b. Once the user completes text entry in a session and changes the application, she is probed to record her perceived emotion during this session. The user provided emotion self-report is associated with this text entry session. The typing and swiping portions are identified from the entire session, relevant

Table 1

Comparison of *TouchSense* with related touch based emotion detection applications in smartphone in terms of (a) type of touch modality (b) usage of additional sensors (c) dependency on specific application and (d) number of detected emotion states. It reveals the scope of investigating the combined influence of touch-based interactions during text entry i.e. *typing*, *swiping* for multiple emotion detection irrespective of target application. ✓ indicates that the modality is used, ✗ indicates that the modality is not used and NA indicates that the modality is Not Applicable for the study.

Reference	Touch modalities						Additional sensors	Application specific?	Inferred emotion	
	Finger-stroke	Tapping	Drag-and-drop	Scaling	Typing	Swiping			Single	Multiple
Gao et al. (2012)	✓	✗	✗	✗	NA	NA	✗	Yes (Fruit Ninja game)		✓
Lee et al. (2012)	NA	NA	NA	NA	✓	✗	✗	Yes (Twitter)		✓
Kim and Choi (2012)	✗	✓	✗	✗	NA	NA	✓	No		✓
Wac et al. (2015)	✓	✗	NA	NA	✓	✓	✗	No	✓	
Ciman and Wac (2016)	✓	✗	NA	NA	✓	✓	✓	No	✓	
Mottelson and Hornbæk (2016)	✗	✓	✓	✓	NA	NA	✓	No	✓	
Shapsough et al. (2016)	NA	NA	NA	NA	✓	✗	✓	No		✓
Tikadar et al. (2017)	✗	✓	✗	✗	NA	NA	✗	Yes (Touch-based game Emotion estimator)		✓
Ghosh et al. (2017a)	NA	NA	NA	NA	✓	✗	✗	No		✓
TapSense Ghosh et al. (2017b)	NA	NA	NA	NA	✓	✗	✗	No		✓
Cao et al. (2017)	NA	NA	NA	NA	✓	✗	✓	No	✓	
<i>TouchSense</i>	NA	NA	NA	NA	✓	✓	✗	No		✓

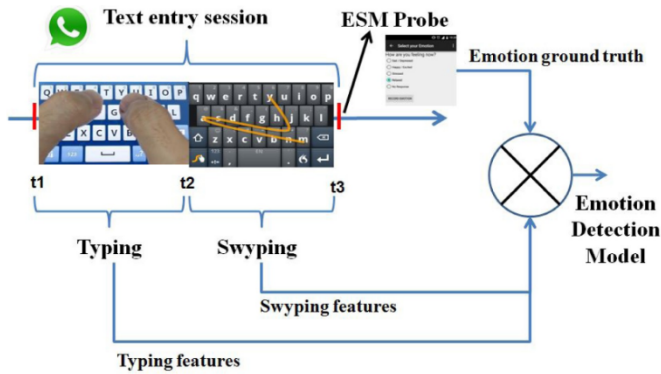
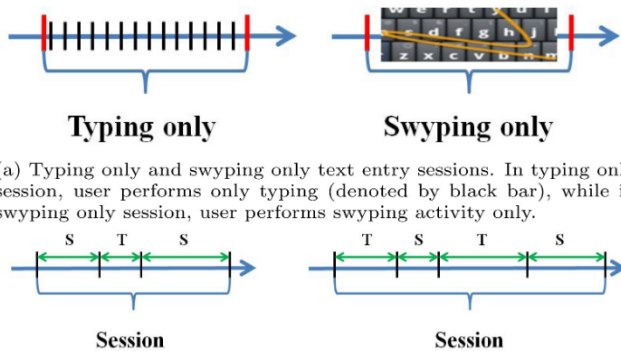


Fig. 2. Scenario of touch based emotion detection during text entry. Elapsed time between t_1 and t_3 is considered a session, when user performs text entry in WhatsApp. Typing ($t_1 - t_2$) and swiping ($t_2 - t_3$) portions are identified, relevant features are extracted from these portions and correlated with the emotion self-report collected via ESM probe to build the emotion detection model.



(a) Typing only and swiping only text entry sessions. In typing only session, user performs only typing (denoted by black bar), while in swiping only session, user performs swiping activity only.

(b) Sample text entry sessions. T, S indicate typing and swiping respectively.

Fig. 3. Example of different types of interaction sessions during text entry.

features are extracted from these portions and correlated with the emotion self-reports to build the emotion detection model.

The scenario described above calls for following design capabilities. (a) Tracing user's text entry activities and discriminate between different touch interactions, such as typing and swiping. Notably, this trace collection should be done in privacy preserving manner. (b)

Collecting self-reports timely from the users with low probing rate to reduce the survey fatigue. (c) Extracting the relevant typing and swiping features from each session and correlate them with emotion self-reports to construct a model for emotion detection. We implement these design choices as an Android application as discussed next.

4. Touchsense implementation

TouchSense is implemented as a client-server application, where the client component runs on the smartphone as an Android app and the server program runs in the background.

We illustrate the architecture of *TouchSense* in Fig. 4. It has the following major components. *TouchLogger*, which traces both type and swipe interactions during text entry. It is implemented by instrumenting QWERT keyboard with Android Input Method Editor (IME) facility (Fig. 5). The keyboard provides all the major functionalities as available in Google keyboard to minimize user inconvenience during text entry. We record the timestamp, associated application name, any non alphanumeric character typed, pressure and speed during every touch interaction. In order to ensure user privacy, we do not store any alphanumeric character. *ESMLogger* triggers the ESM probes to collect the emotion self-reports. We implement the self-report collection UI as shown in Fig. 6. The touch interaction details and the collected self-reports are temporarily stored on the device. Once the user is connected to internet, the log gets uploaded on the server. Once the touch interaction details and self-reports are available on the server, we perform data cleansing operations and carry out feature extraction from raw data log. Later these features are correlated with the emotion self-reports to build the *emotion detection* model. We discuss these steps in detail next.

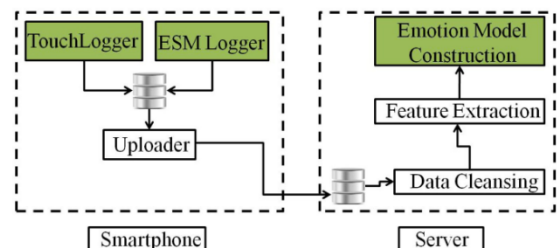


Fig. 4. TouchSense architecture; key application components are highlighted.

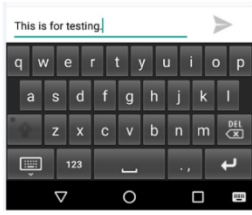


Fig. 5. TouchSense Keyboard.

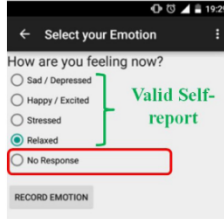


Fig. 6. TouchSense self-report collection interface.

4.1. Typing and swyping session identification

The intuition behind typing and swyping session identification is that there is significant difference in the finger movement speed during typing and swyping interactions. In specific, we measure the touch interaction speed in terms of pixels per second. During swyping, as the user needs to quickly move her finger over multiple letters, the interaction speed is higher; on the contrary, for typing user touches the screen on a single letter and releases the finger, hence there is hardly any finger movement on the surface of the screen resulting in very low interaction speed. We show the same in Fig. 7. We plot the natural logarithm of speed values of typing and swyping interaction and observe the difference in typing and swyping interaction speed.

Based on this finding, we follow the method as outlined in Algorithm 1 to distinguish between typing and swyping interaction in a session. As the user performs text entry in a session, we record the speed at every touch interaction. We define two different thresholds in two steps to discriminate typing and swyping. In the first step, we accumulate all the speed values from all sessions for each user and perform k-means clustering ($K = 2$) (Hartigan and Wong, 1979) to identify two clusters. We use natural logarithm of speed values while doing clustering. We select the cluster with lower centroid and use the maximum value of this cluster as a global threshold (ϕ) (line 2). Any touch event with speed greater than ϕ is characterized as swype instance and vice versa. In the second step, we perform session level analysis to verify if this global threshold (ϕ) needs to be refined for a particular session $session_i$. In case the highest touch speed recorded in $session_i$ is less than ϕ , this is considered a typing only session (line 5 – 7). Similarly, if the lowest speed observed in $session_i$ is higher than ϕ , it is considered as a swype only session (line 8 – 10). Otherwise, $session_i$ is considered as mixed session containing both typing and swyping (line 11 – 14). For a mixed session $session_i$, we again perform k-means clustering ($K = 2$) on the set of speed values recorded in $session_i$ and compute the maximum value of the cluster with low values as session specific threshold ($\phi_s(i)$). Hence, in this mixed session $session_i$, touch interactions with speed value lower than threshold $\phi_s(i)$ is considered as typing and otherwise swyping.

4.2. Self-report collection

Once user completes text entry in a session and switches the application, it is ideal to probe the user. But asking every time, whenever user changes the application after text entry is demanding and fatigue-inducing. So, we implement the LIHF ESM schedule (Ghosh et al.,

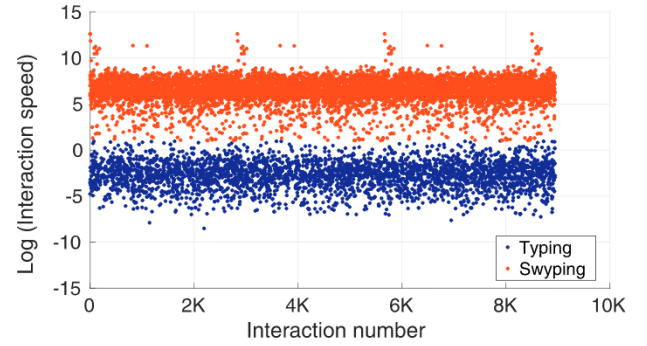


Fig. 7. Comparing interaction speed for typing and swyping activity.

2017c) to relax the probing criteria. The LIHF schedule, as outlined in Fig. 8, probes the user only when (a) she has entered sufficiently long text (L) and (b) a minimum time (W) has elapsed since last probing. We use the values of the schedule parameters as specified in Ghosh et al. (2017c). However, it may be noted that user may have entered text in different applications to record sufficiently long text, which needs to be broken into different sessions (based on application).

We collect the emotion self-reports using survey questionnaire as shown in Fig. 6. We concentrate on four discrete emotion states - *happy*, *sad*, *stressed*, and *relaxed*. The self-report UI design is driven by the following policies

- We select one dominant emotion from each of the four quadrants of the Circumplex model (Russell, 1980) (Fig. 1) so that they are non-overlapping and user can distinguish them well during self-reporting.
- We keep the provision of skipping self-reporting by selecting the *No Response* option. By default, when the UI is displayed, the *No Response* option gets selected. In order to provide the emotion self-report, the user needs to select a valid emotion and record the same.

4.3. Data processing

In the following, we discuss the data cleansing tasks to eliminate the non-operable sessions.

- *Removal of No Response Labels*: We filter out all sessions marked with *No Response* as they do not reveal any emotion.
- *Distant Session Elimination*: Self-report collection may be delayed for different reasons. For example, it may be possible that the user does not notice the probe and locks the screen. If self-report is recorded after a long time since the last interaction in a session, it is less likely to represent the emotion perceived in this session. We remove the sessions where emotion self report is recorded more than 3 h after the last touch interaction in that session.
- *Elimination of Small Sessions*: To ensure that enough information is present either for typing or swyping, so that we could appropriately associate typing and swyping patterns with perceived emotions, we discarded any session with less than 20 touch interactions¹

4.4. Feature selection

In order to classify the emotion states, we compute the features extracted from the touch interactions log during text entry. Primarily, the features can be categorized into two groups - typing and swyping, as summarized in Table 2.

¹ We obtain superior classification performance for session length of 20 interactions, because beyond this the number of sessions reduce, which influence the training performance. So we decide the use minimum session length as 20.

Input: Values of speed from all sessions (V); Session-wise interaction speeds (S), $session_i \in S$, $session_i$ contains the set of interaction speed for i^{th} session

Output: Global threshold speed ϕ , session-wise threshold speed ϕ_s , session-wise labels L

```

/* Compute global threshold
1  $[C_{type}, C_{swipe}] \leftarrow \text{kmeans}((V), 2)$  */
2  $\phi \leftarrow \max(C_{type})$  */
3  $\phi_s \leftarrow []$  /* Having computed a global threshold, compute threshold speed for each session */
4 foreach  $session_i \in S$  do
5   if  $\max(session_i) < \phi$  then */
6     /* Typing only session
7      $\phi_s(i) = \phi$ 
8      $L(i) = \text{typing}$  */
9   else if  $\min(session_i) > \phi$  then */
10    /* Swyping only session
11     $\phi_s(i) = \phi$ 
12     $L(i) = \text{swyping}$  */
13  else
14    /* Combines both typing and swyping
15     $[C_{type}, C_{swipe}] \leftarrow \text{kmeans}((session_i), 2)$ 
16     $\phi_s(i) = \max(C_{type})$ 
17     $L(i) = \text{combined}$  */

```

Algorithm 1. Typing-swyping distinguishing method.

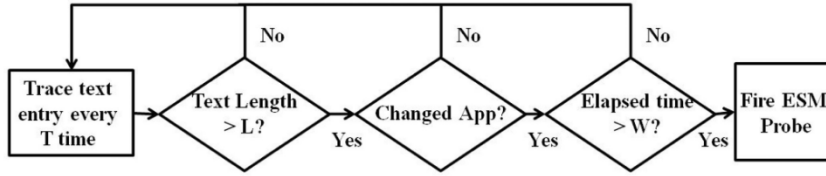


Fig. 8. ESMLogger implements hybrid ESM for collecting emotion labels. We set the values of the config parameters T , L , W as 15 sec, 80 characters and 30 min respectively.

Table 2
Feature Table.

Category	Feature name
Typing	Mean Session ITD (MSI)
	Refined Mean Session ITD (RMSI)
	Number of special characters ($splchar_{per}$)
	Number of backspace (or delete) ($backspace_{per}$)
Swyping	Mean pressure ($pressure_{mean}$)
	Std. dev pressure ($pressure_{sdv}$)
	Mode pressure ($pressure_{mode}$)
	Swype percentage ($swype_{per}$)
	Session duration (duration)
Both (Typing, Swyping)	

4.4.0.1. Typing Features. We consider typing speed as a feature. In a session, we compute the time interval between two consecutive key pressing events, defined as Inter-Tap Duration (ITD). We compute *Mean Session ITD (MSI)*, by taking average of all Inter-Tap Duration (ITDs) present in a session.

However, we observe that it is possible to have overlapping ITD values in two consecutive typing sessions, tagged with different emotion states, if the emotion labels are captured within a short time interval. This may result due to the influence of the previous emotion on the next session. Therefore *mean session ITD (MSI)* computed from all ITD values present in a session may not be very effective in distinguishing two emotions (due the presence of overlapping ITDs, the mean values of two sessions will be close to each other). But if we can identify the non-overlapping set of ITDs from each session and use only those ITDs to compute the mean ITD, the difference in mean values will be pronounced and help in distinguishing two different emotions. Hence, we introduce the feature *RMSI*, which is calculated based on the dominant set of ITDs present in a session. Fig. 9 describes the intuition behind selecting *RMSI* as the feature. We identify the major cluster and compute mean giving it the preference so that the difference in *RMSI* become more pronounced for two different emotion states. We implement the k-means clustering based approach to compute the *RMSI* as outlined in Algorithm 2.

In addition, we consider the following keystroke based features. We compute the fraction of backspace and delete keys typed in a session and use it as a feature $backspace_{per}$. This feature provides an estimation of the number of mistakes being made in a session. Similarly, we compute the fraction of special characters (any non-alphanumeric character) typed in a session as a feature $splchar_{per}$.

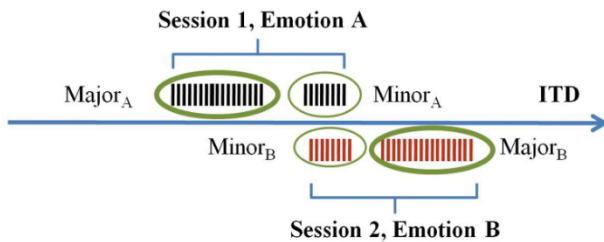


Fig. 9. Schematic showing the intuition behind using *RMSI* as feature. Identification of dominant set of ITDs in a session and giving it preference while computing *RMSI* provides it better distinguishing ability to identify two emotion states.

4.4.0.2. Swyping Features. Once the swyping interactions are identified in a session, we compute the feature $pressure_{mean}$ as the mean of all pressure values obtained from these interactions. Additionally, we compute the standard deviation and mode of swype pressure to obtain the features $pressure_{sdv}$ and $pressure_{mode}$ respectively. The fraction of swype interactions in a session is computed as a feature $swype_{per}$.

Apart from these two sets of features, we also use session duration as a feature. It measures the duration of a session irrespective of its type (typing only, swyping only, or combined).

4.5. Model construction

The model construction is performed on the server side. We build personalized models using the above-mentioned features to classify four emotion states. We implement Random Forests supervised machine learning algorithm in Weka (Hall et al., 2009) to build the emotion detection model.

5. Data collection

In this section, we discuss the field study and provide an overview of the collected dataset.

5.1. Field study

Survey Participants: We recruited 30 university students (25 male, 5 female, aged between 18 – 35 years) to participate in our experiments. We installed *TouchSense* on their smartphones and instructed them to use it for 3 weeks to perform text entry and record their emotion self-reports. 4 participants left the study in the middle and 4 participants entered less than 30 labels during entire period. Finally, we collected data from the remaining 22 users (18 male, 4 female). The average age of the selected set of participants is 23.3 years (std. dev. 4.23). All of them reside in the university campus at IIT Kharagpur (India). At the end of data collection period, each of the participants was given a monetary reward equivalent to 10 USD.

Instructions to the Participants: We instructed the participants to select the *TouchSense Keyboard* as the default keyboard and use the same for their daily text recording activities. We informed the participants that once they switch from an application, which involved typing or swyping, they may receive a survey questionnaire as a pop-up, where they need to report their emotion state. We also advised participants to record *No Response* label if they want to skip self-reporting.

5.2. Data overview

We collected a total of 589,525 touch interactions spanning across 2,929 sessions from the field study. We summarize the final dataset in Table 3 and show the distribution of different types of session for every user in Fig. 10. This reveals that most of the participants use both types of interactions in a session.

We distinguish each touch interaction as typing or swyping following Algorithm 1. In Fig. 11, we show the what percentage of total touch interactions corresponds to typing and swyping for every user. We observe that for each user, the swype fraction is more than 15% except two (U_2 , U_4). However, the maximum swype percentage is found to be 53.76%, with an average of 29.94% (std. dev 13%).

Input: Session ITDs(S)

Output: $RMSI$

```

1  $[C_{major}, C_{minor}] \leftarrow kmeans((S), 2)$ 
2  $itd_{major} \leftarrow mean(C_{major})$ 
3  $itd_{minor} \leftarrow mean(C_{minor})$ 
4  $S_{sorted} \leftarrow sort(S)$  in ascending order
5 if  $itd_{major} < itd_{minor}$  then
6    $RMSI \leftarrow$  Compute mean using top 80% samples of  $S_{sorted}$ 
7 else
8    $RMSI \leftarrow$  Compute mean using bottom 80% samples of  $S_{sorted}$ 

```

Algorithm 2. $RMSI$ calculation method.

Table 3
Final dataset.

Total touch events	589,525
Total sessions	2929
Total duration	68 Hr.
Per user sessions (mean, SD, minimum)	133, 127, 30
Median session duration	153 sec.
Median session length	86

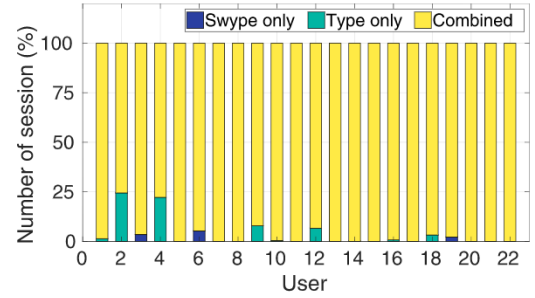


Fig. 10. Comparison of typing only, swiping only and combined sessions reveals that most of the participants use both types of interactions in a session.

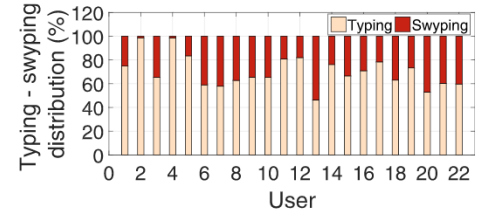


Fig. 11. Distribution of Type and Swype interactions performed by each user.

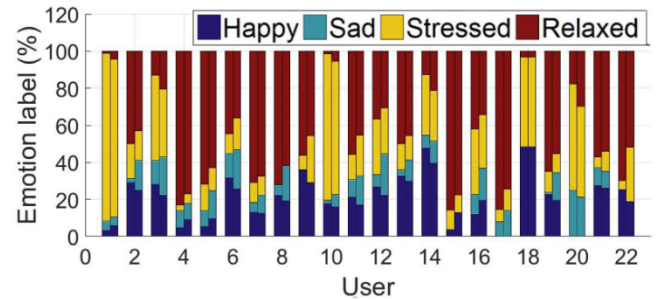


Fig. 12. Emotion distribution of each user. All but 7 users have recorded every emotion state. For every user, there are two bars - the first bar indicates the distribution of emotion samples in original data as recorded by the participants, the corresponding second bar indicates the distribution of emotion samples after over-sampling using SMOTE.

We also exhibit the frequency distribution of different emotion states for each user. It is observed that all but 7 users ($U8$, $U9$, $U15$, $U17$, $U18$, $U20$, $U22$) have recorded four emotion states (Fig. 12). We identify *relaxed* as the most commonly recorded emotion state. Overall, we record 16%, 6%, 38% and 40% sessions tagged with *happy*, *sad*, *stressed*, and *relaxed* respectively, which reveals the class imbalance across different emotion states.

5.3. Reducing class imbalance in emotion distribution

We rely on Synthetic Minority Over-sampling Technique (SMOTE) (Chawla et al., 2002) to address the problem of data imbalance in emotion samples. SMOTE re-samples the class with the least number of instances so that almost all classes are equally balanced.

Table 4

User-wise percentage of newly added samples using SMOTE.

User	Added Sample (%)	User	Added Sample (%)	User	Added Sample (%)
U1	6.59	U9	39.06	U17	14.67
U2	16.27	U10	9.95	U18	0.00
U3	26.02	U11	23.07	U19	16.67
U4	7.78	U12	20.0	U20	17.50
U5	14.13	U13	9.37	U21	5.71
U6	23.68	U14	19.77	U22	34.51
U7	5.26	U15	10.62	-	NA
U8	16.67	U16	22.53	-	NA

While using SMOTE we ensured that - (a) we do not include any new state i.e. if the user has not originally provided any emotion state, the same is not added after sampling and (b) we try to add as few records as possible, so that the number of instances in least represented two emotion states are almost equal. By applying SMOTE we add 14.27% new samples in the dataset. Additional data introduced per user is shown in Table 4. In Fig. 12, we show the comparison of emotion state distribution before and after applying SMOTE for every user.

6. Evaluation

We evaluate the emotion detection model as implemented in *TouchSense* using 10-fold cross validation. We consider AUCROC (Area under the Receiver Operating Characteristic curve) and F-score as the performance metric. We measure how accurately each emotion state is determined using these two metrics. We also report the per user accuracy (AUCROC), using the weighted average of AUCROC (auc_{wt}) from four different emotion states as per Eq. 1, where f_i & auc_i indicate the fraction of samples and AUCROC for emotion state i respectively.

$$auc_{wt} = \sum_{\forall i \in \{happy, sad, stressed, relaxed\}} f_i * auc_i \quad (1)$$

We use AUCROC to measure the classification performance as it is the suitable metric for unbalanced dataset (Bradley, 1997; Tang et al., 2009a). Although we use SMOTE, it balances the two least represented classes, not all the classes are equally represented in the collected data.

6.1. Baseline algorithms

We compare the performance of *TouchSense* with the following baseline emotion models as outlined below.

- **Typing only model (TYP):** We construct a personalized emotion detection model based on typing features only (Table 2). This model is constructed by extracting features from the typing only sessions and typing only interactions of the combined sessions. We use this model to understand the effectiveness of typing only features in emotion detection.
- **Swyping only model (SWP):** This personalized emotion detection model is constructed based on swype features only (Table 2). It is constructed by extracting features from the swyping only sessions

and swyping only interactions present in the combined sessions. We use this model as a baseline to find out the role of swype only features on emotion detection.

- **Aggregate model (AGG):** As *TouchSense* implements a personalized model, it requires individual training. So, we aim to reduce the overhead of personalized training using leave-one-participant-out-cross-validation i.e. we build the model for one user using data from other users and test the model using this user's data. The underlying assumption of this model is that there exists similarity in touch interactions across different users. This model is constructed using the same set of features as used in *TouchSense* (Table 2).
- **Most represented emotion model (MRE):** In our dataset, we observe that for most of the users, there exist one emotion, which dominates the frequency distribution of emotion self-reports (Fig. 12). Following this observation, we develop a personalized model, which always produces the most represented emotion as the output. Comparison with this baseline brings out the benefit of constructing a touch interaction based model over always predicting the most dominant state.

6.2. Classification performance

In this section, we evaluate the performance of *TouchSense* for correctly classifying emotion from smartphone touch interactions. We report the classification results in Fig. 13.

We obtain an average accuracy (auc_{wt}) of 73% (std. dev 6%) and a maximum of 87% as shown in Fig. 13a. In Fig. 13b, we observe that all states have the AUCROC close to 70% while *sad* state has AUCROC greater than 85%. Among all the emotion states, *relaxed* exhibits the highest F-score (close to 65%), on the contrary, *happy* shows comparatively low F-score. We obtain high AUCROC ($> 75\%$) for 25% of the participants, while attain decent performance for most of the users resulting low variability is user-wise AUCROC. Notably, we observe poor performance ($AUCROC \leq 65\%$) for 2 users (*U13*, *U21*). For *U13*, the explored features could not distinguish *happy* and *stressed* states, similarly for *U21*, *happy* and *sad* states are identified less accurately resulting in comparatively poor performance for these two users.

6.2.1. Effect of SMOTE on classification performance

We compare the difference in classification performance for the two cases - original dataset, over-sampled dataset using SMOTE. The average accuracy (auc_{wt}) is 53% for the original dataset, while it is 73% after applying SMOTE. We also compare the average AUCROC for each emotions for both the cases in Table 5. We find that emotion-wise performance is poor for original data, however it improves after over-sampling with 14% records using SMOTE. This shows that the proposed model can attain high classification performance with adequate data. Improvement in the over-sampled dataset can be attributed to adding more samples in underrepresented categories.

6.3. Comparison with baseline models

In Fig. 14, we show that the proposed *TouchSense* model outperforms all the competing baseline models. The personalized model based

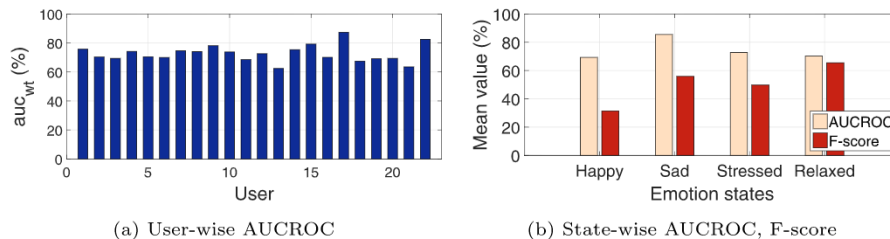


Fig. 13. Emotion classification performance of the proposed personalized (typing and swyping combined) model. We observe user-wise average AUCROC of 73%. All states are identified with AUCROC $\geq 70\%$.

Table 5

Comparing average AUCROC for different emotions on original data and over-sampled data.

Dataset	Happy	Sad	Stressed	Relaxed
Original data	48	50	55	51
Over-sampled data	70	84	71	71

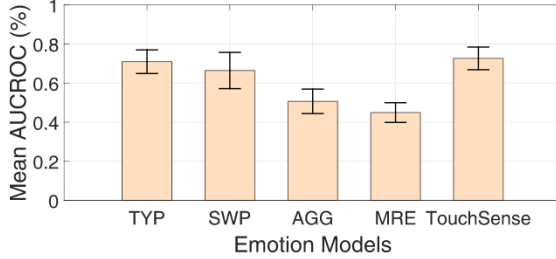


Fig. 14. Mean AUCROC (auc_{wt}) comparison with different models. Error bar indicates standard deviation across user-wise AUCROC (auc_{wt}) values. MRE based model performs worst followed by AGG model. Typing only (TYP) and Swiping only (SWP) model perform moderately, but the combined model used in *TouchSense* outperforms all other models.

on most represented emotion (*MRE*) exhibits significantly poor mean AUCROC of 44% (std. dev 5%). Hence, always predicting the dominant emotion is not a good choice. The aggregate model (*AGG*) also performs poorly with an average AUCROC of 51%. This points to the variability in individual typing and swiping behavior, which aggregate model fails to capture. On the contrary, *SWP* and *TYP* models show reasonably better performance, with mean AUCROC of 66% and 71% respectively. Hence, an in depth investigation of *SWP*, *TYP* and *TouchSense* model follows.

6.3.1. Model selection

Close inspection reveals that a majority (55%) of the participants exhibit best classification performance for *TouchSense*, nevertheless *SWP*, *TYP* models show best classification performance for 18% and 27% of participants respectively. We investigate the role of the touch interactions on the performance of the *SWP*, *TYP* and *TouchSense* models for the individual users. We approach this investigation from two perspectives - (a) number of sessions and (b) volume of typing and swiping done by the users. Fig. 15 depicts the average number of text entry sessions for the users, for whom the corresponding model provides the best classification performance. Interestingly, we observe that for users with few number of sessions, *SWP* model performs best. This indicates that when users perform large amount of swiping interactions, considering swiping related features help to improve emotion

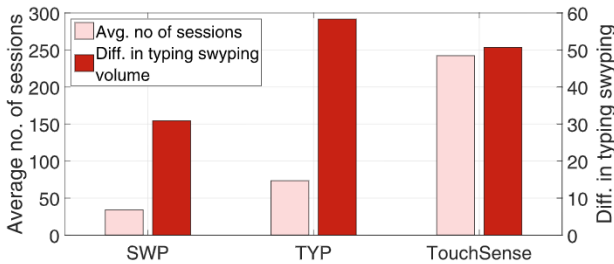


Fig. 15. Comparing average number of sessions, difference in typing and swiping volume for the users having highest classification performance in the corresponding model. On average, *SWP* model has minimum number of sessions followed by *TYP* model followed by combined model in *TouchSense*. However, *TYP* model performs best if the difference between typing and swiping is highest; *SWP* model performs best if the difference is least and the combined model performs best, when the difference is in between.

detection performance. *TYP* model performs best for users with comparatively more sessions and *TouchSense* performs best if the number of sessions are highest. This difference in average number of sessions for each model is also found statistically significant ($p < .05$) using one-way ANOVA test and follow-up Tukey HSD test (for every pair of model).

Next, we investigate the influence of typing and swiping volume on the model performance. In Fig. 15, we also show the difference in the volume of typing and swiping volume among all users for whom the respective classification algorithm performs the best. We observe that for the users exhibiting best performance for *SWP* model, the difference in typing and swiping volume is least ($\approx 30\%$). On the contrary, this difference is highest ($\approx 58\%$) for the users preferring *TYP* model. In general, when the difference between typing and swiping is close to 50% then the classification performance tend to be the best.

6.4. Feature analysis

We measure the effectiveness of different features using information gain (IG) by invoking *InfoGainAttributeEval* method of WEKA (Hall et al., 2009). We show the average information gain and the respective ranking of the features in Table 6. The feature evaluation used 10-fold cross validation. This is evident that $pressure_{mean}$ is the most discriminating feature followed by $pressure_{sdv}$ and *RMSI*. This indicates that among swiping features, $pressure_{mean}$ and among typing features, *RMSI* exhibit the highest discriminating power.

We delve deep to examine the statistical significance of each feature on the emotion classification. For every user, we consider one feature at a time and perform one-way ANOVA test (Salkind, 2010). We form separate group for every emotion. We group together the feature values having same emotion state and observe significant ($p < .05$) difference across emotion states. Next, users exhibiting at least one significantly different emotion state are analyzed further by Tukey HSD test (Lane and Salkin, 2010) for emotion pair-wise significance. In Table 7, we show the fraction of users, for whom each individual feature plays statistically significant role ($p < .05$) in emotion classification. We observe that for most of the users *MSI*, *RMSI*, $splchar_{per}$, $pressure_{mode}$, $pressure_{mean}$ play a crucial role, followed by other features like $backspace_{per}$, *duration*.

6.5. Training duration

In order to deploy in practice, this is necessary that *TouchSense* attains a reasonable accuracy within a short training period. In Fig. 16, we demonstrate the change in classification accuracy (AUCROC) with training period.

We accumulate the data at an interval of every 3 days and measure the AUCROC of *TouchSense* for each user. As expected, the mean AUCROC over all users increases with longer training period and the variation in AUCROC across users also reduces. For instance, within a period of 15 days, an average AUCROC of $\approx 60\%$ is obtained, which attains 72% after 18 days. We also plot the AUCROC of two

Table 6
Discriminating features based on Information Gain.

Feature Name	Rank	Average IG
$pressure_{mean}$	1	0.3163
$pressure_{sdv}$	2	0.2663
<i>RMSI</i>	3	0.2631
$pressure_{mode}$	4	0.2476
$splchar_{per}$	5	0.2452
<i>MSI</i>	6	0.2349
$backspace_{per}$	7	0.2306
$swipe_{per}$	8	0.2299
<i>duration</i>	9	0.0693

Table 7

User coverage using one-way ANOVA. Each cell indicates percentage of users having significantly ($p < .05$) different value for at least one emotion for corresponding feature.

Feature	pressure _{mean}	pressure _{sdv}	RMSI	pressure _{mode}	sp1char _{per}
Users (%)	27.27	18.18	36.36	31.82	36.36
Feature	MSI	backspace _{per}	swipe _{per}	duration	-
Users (%)	54.55	13.64	27.27	13.64	-

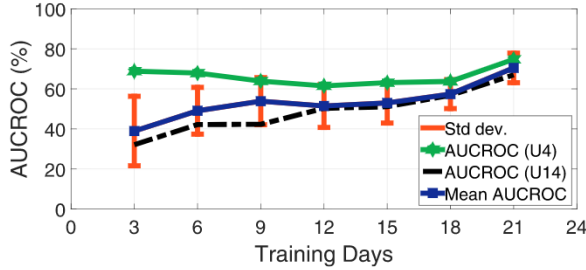


Fig. 16. Classification accuracy (AUCROC) with varying training period. Mean AUCROC improves with time and the std. dev reduces indicating less variability in AUCROC values across different users. For user like U4, AUCROC does not vary much with time, however for most of the users like U14 AUCROC improves with time.

representative users (U4, U14) with varying training period. We observe that for user U4, the accuracy saturates quickly and does not change much with time, whereas for user U14 the accuracy slowly improves with additional training data.

6.6. Post-study quantitative assessment

We conducted a post-study survey following the *Post-Study System Usability Questionnaire* (PSSUQ) (Lewis, 1991) to gauge the effectiveness of *TouchSense*. We asked the participants different questions from the system usability perspective and obtained the rating in a scale of 1 (worst) to 5 (best). We compute the weighted average score as per Eq. 2, where i denotes the rating provided by the user, n_i indicates the number of users provided the rating i and N indicates the total number of users.

$$m_{wt} = \frac{\sum_{i=1}^5 i * n_i}{N} \quad (2)$$

We compute the usability scores based on Eq. 2 and report the same in Fig. 17. We asked (a) if the participants are able to use *TouchSense* easily, to which 78% participants provided a score above 4 and we obtain an average score of 4.14 (*usability*). (b) In terms of easiness of using the *TouchSense* keyboard for typing and swyping activity, we obtain an

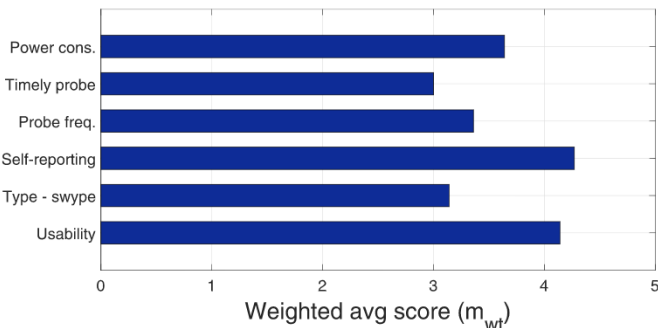


Fig. 17. Average score of different usability parameters based on post-study survey.

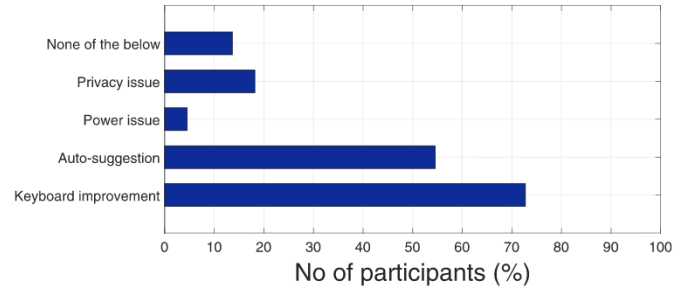


Fig. 18. Source of inconvenience as obtained from the *TouchSense* post-study survey.

average score of 3.14 (*type-swype*). (c) We also asked if the participant can easily distinguish between four emotion states and record her emotion while self-reporting, to which 86% participants provide a score greater than 4, which finally results in mean score of 4.27 (*self-reporting*). (d) We measure the user engagement in terms of probing frequency, probing at opportune moments and energy consumption. 55% and 41% participants expressed satisfaction in terms of the probing rate and opportune probing time respectively. Precisely, we obtain an average score of 3.64 (*probe freq*) and 3 (*timely probe*) respectively from these two aspects. Finally, we survey if the battery drainage is significant for *TouchSense*, for which 86% participants disagreed and we obtain an average score of 3.6 (*power cons*) in this regard.

In this quantitative study, we also identified the source of inconveniences while using *TouchSense*. We summarize the findings in Fig. 18. (a) We observe that the major source of inconvenience is the *TouchSense* keyboard as 72% of participants indicated keyboard needs to be improved. This is reasonable, as all the participants are accustomed with the standard Google keyboard. For instance, users U1, U7 suggested to increase the space between two keys. (b) Another source of inconvenience was the quality of auto-suggestion as indicated by the 54% participants. In *TouchSense* we implement a naive dictionary based algorithm to suggest words, which is not as sophisticated as commercial applications. (c) Since we explained to our participants apriori that only typing metadata will be stored, not the content, this assured them regarding privacy; only 18% participants were concerned about privacy. (d) Energy consumption also do not seem to be a concern for most of the participants. In summary, our usability study indicates the possibility of deploying *TouchSense* as a non-intrusive, low-power emotion detection application.

7. Discussion

The key finding of this work is that joint influence of touch interactions during text entry is effective in determining multiple emotion states. At the same time, we learn different insights, which are important to consider while deploying a touch based emotion detection application like *TouchSense*. We discuss those and the limitations of the study next.

First, the model performance heavily depends on individual user's training data. However, we observe that due to in-the-wild nature of the study it may not be possible to collect a balanced dataset. Although, we use SMOTE to overcome the sample imbalance issue, this should be considered carefully to obtain balanced dataset while deploying such an application. It may be also possible to use machine learning algorithms especially designed for unbalanced dataset (Breiman et al., 2004; Tang et al., 2009b). Second, we observe that for few users based on the amount of typing and swyping, the typing-only or swyping-only model returns best performance (Section 6.3.1). So, this should be taken into account to obtain best classification performance from the model. Finally, we eliminate small sessions where the amount of entered text is not adequate. This essentially increases the data collection period, which should also be considered while designing the study.

One of the limitations of this work is that we assume user emotion does not change within a session. Although in our dataset, we observe that 90% of the sessions are having duration less than 3 min, indicating this assumption unlikely to be false, but this may not hold true always. Designing advanced ESM schedule, which can trace in-session emotion variation this problem can be mitigated. However, this has the risk of high survey fatigue. Another possible limitation of this work is keyboard (as indicated in user study), which can be improved for better user experience and smooth data collection. We have used custom questionnaires to obtain post-study participant feedback. However, this can be improved by using usability evaluation questionnaires like USE, SUS (Brooke et al., 1996; Lund, 2001). Finally, the emotion detection performance can be improved by considering additional modality like accelerometer, gyroscope.

8. Conclusion

This paper investigates the role of different touch interactions during text entry in smartphone for multi-state emotion detection. We design, implement, and deploy an Android application *TouchSense*, which records touch interactions using soft keyboard in smartphone, and collects self-reported emotion states using an Experience Sampling Method (ESM). It distinguishes between the touch interactions i.e. typing and swiping and extracts features corresponding to individual touch interactions. We correlate both typing and swiping related features with self-reported emotion labels to devise a personalized Random Forest based emotion detection model to identify four emotion states (*happy*, *sad*, *stressed* and *relaxed*). We obtain an average accuracy (AUCROC) of 73% (std dev. 6%) and a maximum of 87% in a 3-week in-the-wild study. Analyzing the role of different features, we find that the mean pressure during swiping and proposed representation of typing speed play a major role in distinguishing multiple emotion states. We also highlight that if there is large number of text entry sessions and there is high difference in typing and swiping volume, both typing and swiping features are to be combined for better classification performance. The post-study participant survey indicates the potential of *TouchSense* as non-intrusive touch based emotion detection application suitable for long-running user studies.

Declarations of interest

None.

References

- Andjelkovic, I., Parra, D., O'Donovan, J., 2018. Moodplay: interactive music recommendation based on artists mood similarity. *Int. J. Human-Comput. Stud.* 121, 142–159.
- Bogomolov, A., Lepri, B., Ferron, M., Pianesi, F., Pentland, A.S., 2014. Daily stress recognition from mobile phone data, weather conditions and individual traits. *Proceedings of the 22nd ACM International Conference on Multimedia*.
- Bogomolov, A., Lepri, B., Pianesi, F., 2013. Happiness recognition from mobile phone data. *Proceedings of the IEEE International Conference on Social Computing (SocialCom)*.
- Bradley, A.P., 1997. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Patt. Recogn.* 30 (7), 1145–1159.
- Breiman, L., Chen, C., Liaw, A., 2004. Using random forest to learn imbalanced data. *J. Mach. Learn. Res.* 6(66).
- Brooke, J., et al., 1996. Sus-a quick and dirty usability scale. *Usab. Evaluat Indus.* 189 (194), 4–7.
- Cao, B., Zheng, L., Zhang, C., Yu, P.S., Piscitello, A., Zulueta, J., Ajilore, O., Ryan, K., Leow, A.D., 2017. Deepmood: Modeling mobile phone typing dynamics for mood detection. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM*, pp. 747–755.
- Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. Smote: synthetic minority over-sampling technique. *J. Artif. Intell. Res.* 16, 321–357.
- Ciman, M., Wac, K., 2016. Individuals' stress assessment using human-smartphone interaction analysis. *IEEE Trans. Affect. Comput. PP* (99). <https://doi.org/10.1109/TAFFC.2016.2592504>, 1–1.
- Consolvo, S., Walker, M., 2003. Using the experience sampling method to evaluate ubi-comp applications. *IEEE Pervas. Comput.* 2 (2), 24–31.
- Epp, C., Lippold, M., Mandryk, R.L., 2011. Identifying emotional states using keystroke dynamics. *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pp. 715–724.
- Gao, Y., Bianchi-Berthouze, N., Meng, H., 2012. What does touch tell us about emotions in touchscreen-based gameplay? *ACM Trans. Comput. Human Interact.* 19 (4).
- Ghosh, S., Ganguly, N., Mitra, B., De, P., 2017a. Evaluating effectiveness of smartphone typing as an indicator of user emotion. *Affective Computing and Intelligent Interaction (ACII)*, 2017 Seventh International Conference on. IEEE, pp. 146–151.
- Ghosh, S., Ganguly, N., Mitra, B., De, P., 2017b. Tapsense: Combining self-report patterns and typing characteristics for smartphone based emotion detection. *ACM MobileHCI*.
- Ghosh, S., Ganguly, N., Mitra, B., De, P., 2017c. Towards designing an intelligent experience sampling method for emotion detection. *Proceedings of the IEEE CCNC*.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., 2009. The weka data mining software: an update. *SIGKDD Explor. Newsl.* 11 (1), 10–18.
- Hartigan, J.A., Wong, M.A., 1979. Algorithm as 136: a k-means clustering algorithm. *J. Roy. Statist. Soc. Series C (Appl. Statist.)* 28 (1), 100–108.
- Hektner, J.M., Schmidt, J.A., Csikszentmihalyi, M., 2007. *Experience Sampling Method: Measuring the Quality of Everyday Life*. Sage.
- Jiang, F., Zarepour, E., Hassan, M., Seneviratne, A., Mohapatra, P., 2015. When to type, talk, or swipe: Characterizing energy consumption of mobile input modalities. *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 114–122. <https://doi.org/10.1109/PERCOM.2015.7146517>.
- Kim, H.-J., Choi, Y.S., 2012. Exploring emotional preference for smartphone applications. *IEEE Consumer Communications and Networking Conference (CCNC)*.
- Kolakowska, A., 2013. A review of emotion recognition methods based on keystroke dynamics and mouse movements. *Proceedings of the 6th IEEE International Conference on Human System Interaction (HSI)*, pp. 548–555.
- Lam, K.-Y., Wang, J., Ng, J.K.-Y., Han, S., Zheng, L., Kam, C.H.C., Zhu, C.J., 2015. Smartmood: toward pervasive mood tracking and analysis for manic episode detection. *IEEE Trans. Human-Mach. Syst.* 45 (1), 126–131.
- Lane, D., Salkin, N., 2010. Tukey's honestly significant difference (hsd). Salkin, N., *Encyclopedia of Research Design* icinde, Thousand Oaks, CA: SAGE Publications 1, 1566–1571.
- Lee, H., Choi, Y.S., Lee, S., Park, I., 2012. Towards unobtrusive emotion recognition for affective social communication. *IEEE Consumer Communications and Networking Conference (CCNC)*.
- Lee, U., Lee, J., Ko, M., Lee, C., Kim, Y., Yang, S., Yatani, K., Gweon, G., Chung, K.-M., Song, J., 2014. Hooked on smartphones: an exploratory study on smartphone overuse among college students. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pp. 2327–2336.
- Lewis, J.R., 1991. Psychometric evaluation of an after-scenario questionnaire for computer usability studies: the asq. *ACM SIGCHI Bull.* 23 (1), 78–81.
- LiKamWa, R., Liu, Y., Lane, N.D., Zhong, L., 2013. Moodscape: Building a mood sensor from smartphone usage patterns. *Proceeding of the 11th ACM Mobisys*.
- Lu, H., Frauendorfer, D., Rabbi, M., Mast, M.S., Chittaranjan, G.T., Campbell, A.T., Gatica-Perez, D., Choudhury, T., 2012. Stressense: Detecting stress in unconstrained acoustic environments using smartphones. *Proceedings of ACM UbiComp*.
- Lund, A.M., 2001. Measuring usability with the use questionnaire12. *Usab. interf.* 8 (2), 3–6.
- McGaugh, J.L., 2016. *Emotions and Bodily Responses: aPsychophysiological Approach*. Academic Press.
- Mottelson, A., Hornbæk, K., 2016. An affect detection technique using mobile commodity sensors in the wild. *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing. ACM*, pp. 781–792.
- Pielot, M., Dingler, T., Pedro, J.S., Oliver, N., 2015. When attention is not scarce-detecting boredom from mobile phone usage. *Proceedings of the ACM UbiComp*.
- Politou, E., Alepis, E., Patsakis, C., 2017. A survey on mobile affective computing. *Computer Science Review* 25, 79–100.
- Rachuri, K.K., Musolesi, M., Mascolo, C., Rentfrow, P.J., Longworth, C., Aucinas, A., 2010. Emotionsense: A mobile phones based adaptive platform for experimental social psychology research. *Proceedings of ACM UbiComp*.
- Roshanaei, M., Han, R., Mishra, S., 2017. Emotionsensing: Predicting mobile user emotion. *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017. ACM*, pp. 325–330.
- Russell, J.A., 1980. A circumplex model of affect. *J. Personal. Social Psychol.* 39 (6), 1161–1178.
- Salkin, N.J., 2010. *Encyclopedia of Research Design*. 1 Sage.
- Sano, A., Picard, R.W., 2013. Stress recognition using wearable sensors and mobile phones. *Affective Computing and Intelligent Interaction (ACII)*, Humaine Association Conference on. IEEE, pp. 671–676.
- Shapsough, S., Hesham, A., Elkhazraty, Y., Zualkernan, I.A., Aloul, F., 2016. Emotion recognition using mobile phones. *e-Health Networking, Applications and Services (Healthcom)*, 2016 IEEE 18th International Conference on. IEEE, pp. 1–6.
- Tang, Y., Zhang, Y.-Q., Chawla, N.V., Krasser, S., 2009a. Svms modeling for highly imbalanced classification. *IEEE Trans. Syst., Man, Cybernet., Part B (Cybernet.)* 39 (1), 281–288.
- Tang, Y., Zhang, Y.-Q., Chawla, N.V., Krasser, S., 2009a. Svms modeling for highly imbalanced classification. *IEEE Trans. Syst., Man, Cybernet., Part B (Cybernet.)* 39 (1), 281–288.
- Tikadar, S., Kazipeta, S., Ganji, C., Bhattacharya, S., 2017. A minimalist approach for identifying affective states for mobile interaction design. *IFIP Conference on Human-Computer Interaction. Springer*, pp. 3–12.
- Trojan, M., Arndt, F., Weinmann, M., Ortmeyer, F., 2013. Emotion recognition through keystroke dynamics on touchscreen keyboards. *ICEIS* (3), pp. 31–37.
- Wac, K., Ciman, M., Gaggi, O., 2015. isensestress: Assessing stress through human-smartphone interaction analysis. *9th International Conference on Pervasive Computing Technologies for Healthcare-PervasiveHealth*, pp. 8–p.