Poster: Effectiveness of Deep Neural Network Model in Typing-based Emotion Detection on Smartphones

Surjya Ghosh
IIT Kharagpur, India
surjya.ghosh@iitkgp.ac.in

Niloy Ganguly
IIT Kharagpur, India
niloy@cse.iitkgp.ernet.in

Bivas Mitra
IIT Kharagpur, India
bivas@cse.iitkgp.ernet.in

Pradipta De
Georgia Southern University, USA
pde@georgiasouthern.edu

ABSTRACT

Typing characteristics on smartphones can provide clues for emotion detection. Collecting large volumes of typing data is also easy on smartphones. This motivates the use of Deep Neural Network (DNN) to determine emotion states from smartphone typing. In this work, we developed a DNN model based on typing features to predict four emotion states (happy, sad, stressed, relaxed) and investigate its performance on a smartphone. The evaluation of the model in a 3-week study with 15 participants reveals that it can reliably detect emotions with an average accuracy of 80% with peak CPU utilization less than 15%.

CCS CONCEPTS

• **Human-centered computing → Human computer interaction (HCI)**; *Smartphones*;

KEYWORDS

Emotion detection; Typing; Deep neural network; Smartphone

ACM Reference Format:
Surjya Ghosh, Niloy Ganguly, Bivas Mitra, and Pradipta De. 2018. Poster: Effectiveness of Deep Neural Network Model in Typing-based Emotion Detection on Smartphones. In *The 24th Annual International Conference on Mobile Computing and Networking (MobiCom '18), October 29-November 2, 2018, New Delhi, India.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3241539.3267761

MobiCom '18, October 29-November 2, 2018, New Delhi, India

ACM ISBN 978-1-4503-5903-0/18/10.
https://doi.org/10.1145/3241539.3267761

1 INTRODUCTION

The ubiquitous presence of smartphones in our daily life makes it a suitable platform for inferring human emotion [5]. Of different activities performed on a smartphone, text input is a significant proportion, as well as, carries emotion signatures. Collecting typing data is also low overhead in terms of energy, thus encouraging investigations of typing based emotion detection on smartphones [2, 3]. Deep Neural Networks (DNN) has come up as an effective tool in identifying patterns, such as in facial expression detection, image recognition, health monitoring [1, 4]. We pose the question whether it is suitable for emotion recognition from typing on smartphones?

DNN models require large volume of data for training. As instant messaging apps are used frequently, it provides the opportunity to collect large data. In DeepMood, Cao et al. could determine bipolar disorder using typing interactions on an end-to-end deep architecture [1]. We explore if DNN models can be trained to determine multiple emotion states, *happy, sad, stressed, relaxed*, based on typing on smartphones.

We develop a DNN model based on Multilayer Perceptron (MLP) using typing features like speed, typing mistakes, special characters and deploy the same on the phone to determine the emotion state. We gather typing data from 15 participants using a custom keyboard and record their self-reported emotion states in a 3-week study. Our key results demonstrate that proposed deep learning model can detect the emotion states with an average accuracy of 80% (std dev. 7.1%). It also reveals that inferring emotion on smartphone based on this model is not resource-intensive (peak CPU utilization: 15%, model size: 3.7 KB, model load time: 240 ms, inference time: 3.2 ms).

2 METHODOLOGY

We develop a custom QWERTY keyboard to trace user's typing activities. It is implemented using the Android Input Method Editor (IME) facility. We identify typing *session*,

defined as the time period spent by the user in a single application without changing the same, as users perform text entry. We extract different typing features like typing speed, typing errors, typing duration from a session. To ensure user privacy, we do not store any alphanumeric character.

Once user completes typing in a session, she is probed to record her emotion during the session. The user is provided with the option to record any of the four emotions - *happy, sad, stressed, relaxed*. These emotions are non-overlapping and are selected from four different quadrants of emotion Circumplex model [6] so that the user can distinguish them well during self-reporting. The self-reported emotion labels collected after typing sessions are used as ground truth. These are correlated with the features extracted from the corresponding session to construct the emotion detection model.

2.1 Emotion Detection DNN Model

We develop the emotion detection model using Multilayer Perceptron (MLP). It implements two hidden layers. The input layer takes 16-dimensional feature input and passes it through the first hidden layer to obtain a 16-dimensional output. The second hidden layer takes this as input and produces a 8-dimensional output, which is fed to the output layer. It produces 4-dimensional output corresponding to every emotion. We use softmax activation with cross entropy loss for classification. Dropout is used for regularization. We use a dropout rate of 0.2 and batch size as 8.

Feature name	Feature description
MSI (Mean session ITD)	Avg. of all ITDs in the session
RMSI (Refined MSI)	Avg. of non-outlier ITDs in the session
Itd^i_{per} , $i \in \{25, 50, 75, 90\}$	i^{th} percentile value of ITDs in the session
Mean_word_time	Average time to complete a word
Std_word_time	Std dev. of word completion times
Session_dur	Duration of the session
Pause_time	Sum of ITDs greater than 30 secs.
No_pauses	No of ITDs greater than 30 secs.
Abs_session_dur	Session_dur - Pause_time
Dur_per_char	Session_dur / No. of chars in a session
Dur_per_word	Session_dur / No. of words in a session
Backsp_{per}	% of backspace in the session
Splchar_{per}	% of non-alphanumerics in the session

Table 1: Set of features used to construct the typing based DNN model for emotion detection

We develop personalized model for each user as individual typing patterns vary [3]. We extract typing features (Table 1) from typing sessions and associate emotion self-reports as described in Figure 1. In the figure, elapsed time between `t1` and `t2` is a session, when user performs text entry in WhatsApp uninterruptedly. Each small bar within this session is a key press event and the elapsed time between two subsequent key press is defined as the Inter-Tap Distance (ITD).

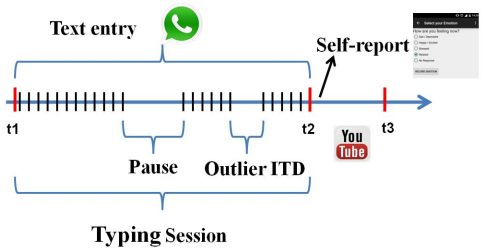


Figure 1: Schematic of feature extraction and self-report correlation with a typing session

The mean of all ITDs represents MSI, while the same of non-outlier (within ± 3 times std dev away from mean) represents RMSI. Any ITD value greater than 30 seconds is considered a pause. The sum of all such ITDs in a session represents pause time. While the total elapsed time between `t1` and `t2` indicates session duration, subtracting the pause time from this, we obtain absolute session duration (Abs_session_dur). The time required to complete a word in a session is obtained by summing up all ITDs used to record a word (space bar indicates completion of a word). The mean and std dev. of word entry times are computed accordingly. The percentage of backspace and special characters used in a session are also used as features. These features and emotion self-reports are fed to the input layer of the DNN to build the emotion detection model.

3 FIELD STUDY & DATASET

We installed the app in the smartphones of 15 students (12 male, 3 female, aged between 24 - 33 years) and collected 3-week typing and emotion self-reports data. The participants were instructed to use the keyboard for typing and emotion reporting when the survey pop-up appears.

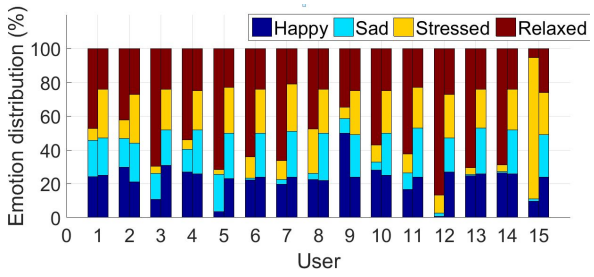


Figure 2: Distribution of emotion self-reports for every user. For every user, first bar shows the distribution in original data and the second one shows the distribution after applying SMOTE. All emotions are almost equally distributed after applying SMOTE.

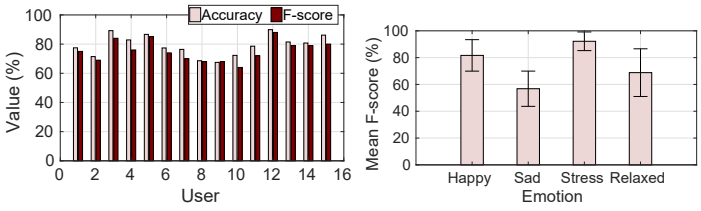
The collected dataset is skewed as users often tend to report *relaxed* emotion, as observed in other uncontrolled collection of emotion self-reports also [5]. We overcome data

imbalance by applying Synthetic Minority Over-sampling Technique (SMOTE). SMOTE is designed to re-sample the class (emotion state) with the least number of instances so that all classes have almost equal samples. After balancing the dataset, we have in total 8301 typing sessions (on average 553 sessions per user, std. dev. 504.2). We show the distribution of these typing sessions tagged with different emotion labels before and after applying SMOTE in Figure 2.

4 EVALUATION

4.1 Model Performance

We perform 10-fold cross validation to evaluate the model. We report the user-wise classification accuracy and F-score in Figure 3a. We obtain an average accuracy of 80% (std dev. 7.1%). The minimum accuracy obtained across all users is 69%, while the highest one is 90%. We obtain an average F-score of 75% (std dev. 7%). We also report the state-wise F-score in Figure 3b. It is observed that all emotions except *sad* are having a F-score more than 70%, while *stressed* is detected with an F-score of 92%.



(a) User-wise Accuracy, F-score **(b) Emotion-wise F-score**

Figure 3: Emotion classification performance of the proposed DNN model. Error bar indicates standard deviation.

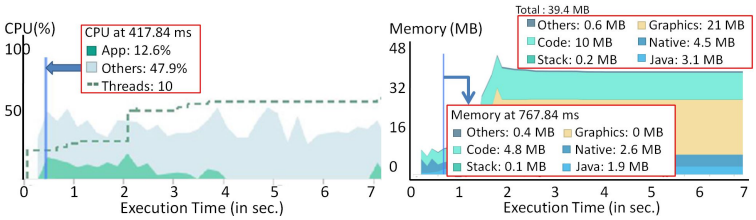
4.2 Resource Overhead

In order to measure resource overhead, we deploy the models on smartphone. We use Tensorflow to generate the deployable model for Android system. We deploy the models on a Moto G2 phone. We show the CPU utilization and memory consumption of the app in Figure 4a, 4b respectively. It is observed that peak CPU utilization is less than 15%, whereas the cumulative memory consumption is less than 40 MB.

Parameter	Mean	Std dev
Model size (in KB)	3.7	0.0
Load time (in msec.)	240.5	12.6
Inference time (in msec.)	3.2	9.6

Table 2: User-wise mean and std deviation of trained model size, model load time and inference time

We also measure trained model size and time required to load the model file for every user. We compute the average model size and average load time for every user and report the same in Table 2.



(a) CPU utilization **(b) Memory consumption**

Figure 4: Resource overhead in terms of CPU utilization and memory consumption (peak CPU utilization less than 15% and cumulative memory consumption less than 40 MB)

Similarly, we compute the average inference time for a test instance. We randomly select 20% samples from every user and use corresponding model to predict the outcome. The time required to infer the emotion for a single test instance is considered as inference time. We note the average inference time for all test instances from every user in Table 2 also. We observe a high standard deviation in inference time. This is primarily because that first inference usually takes more time, later on once the model file is loaded, the inference time is reduced by large amount, thus resulting in high standard deviation.

5 CONCLUSION

In this paper, we investigate the feasibility of executing DNN models on smartphone to determine multiple emotion states based on typing. We develop a MLP based personalized DNN model based on typing features to determine four emotion states (*happy, sad, stressed, relaxed*) and deploy the same on smartphone. The evaluation of the model reveals that it is possible to determine these states with an average accuracy of 80% without major resource overhead.

REFERENCES

[1] Bokai Cao, Lei Zheng, Chenwei Zhang, Philip S Yu, Andrea Piscitello, John Zulueta, Olu Ajilore, Kelly Ryan, and Alex D Leow. 2017. Deepmood: modeling mobile phone typing dynamics for mood detection. In *ACM SIGKDD*.

[2] Surjya Ghosh, Niloy Ganguly, Bivas Mitra, and Pradipta De. 2017a. Evaluating effectiveness of smartphone typing as an indicator of user emotion. In *IEEE ACII*.

[3] Surjya Ghosh, Niloy Ganguly, Bivas Mitra, and Pradipta De. 2017b. TapSense: Combining Self-Report Patterns and Typing Characteristics for Smartphone based Emotion Detection. In *ACM MobileHCI*.

[4] Natasha Jaques, Sara Taylor, Akane Sano, Rosalind Picard, and others. 2017. Predicting tomorrow's mood, health, and stress level using personalized multitask learning and domain adaptation. In *IJCAI 2017 Workshop on Artificial Intelligence in Affective Computing*. 17–33.

[5] Robert LiKamWa, Yunxin Liu, Nicholas D Lane, and Lin Zhong. 2013. Moodscope: Building a mood sensor from smartphone usage patterns. In *ACM Mobisys*.

[6] James A Russell. 1980. A circumplex model of affect. *Journal of Personality and Social Psychology* 39, 6 (1980), 1161–1178.