Computers and Electrical Engineering, vol. 60, pp. 1-13, 2017

Emotion Recognition Using Mobile Phones

I. Zualkernan, F. Aloul, S. Shapsough, A. Hesham, and Y. El-Khorzaty
Department of Computer Science and Engineering
American University of Sharjah, UAE

Abstract – The availability of built-in sensors in mobile phones has enabled a host of innovative applications. One class of application deals with detecting a user's emotions. Previous applications have primarily relied on recording and displaying self-reported emotions. This paper presents an intelligent emotion detection system for mobile phones implemented as a smart keyboard that infers a user's emotional state using machine learning techniques. The system uses accelerometer readings and various aspect of typing behavior like speed and delay between letters to train a classifier to predict emotions. Naïve Bayes, J48, IBK, Multi-response linear regression and SVM were evaluated and J48 was found to be the best classifier with over 90% accuracy and precision. In addition to providing emotive feedback to individual users, the system also uses geo-tagged data to collect and display emotional states of regions or countries through a website.

Keywords – Emotion recognition, Machine intelligence, Mobile phones, Sensors

1. INTRODUCTION

WITH the advent of computing came a growing dependency on smartphones that went beyond the communication purpose they were originally intended for. People today use mobile phones to carry out a range of daily tasks like shopping, ordering food, etc. In addition, mobile phones are also being used as entertainment hubs. Over time, mobile phones have increasingly become more complex to meet consumer's demands and to satisfy an ever-growing need for more computational power. An average mobile phone now comes equipped with communication modules (Bluetooth, Wi-Fi etc.), an array of sensors (accelerometers, gyroscopes, temperature sensors etc.) and significant computational power. These built-in sensors can be used to deploy unique applications that were not possible in the past.

One area where sensors can be used is to perceive a user's emotional state [1]. By capturing a user's current emotions, a device could intelligently personalize the user's experience. Such technology could support application in many domains such as social media, healthcare, etc. Social networks, such as Facebook and Twitter, would be able to respond differently to users based on their current emotional state. This could allow social networks, for example, to block a user from accessing their services, or send them help if they were in a severely distressed state. Another application in social media can be immediate feedback. A post on twitter or Facebook can be automatically flagged if the majority of viewers responded negatively to it. Another area of application is healthcare where users can keep track of their own psychological health. The application enables them to determine, for instance, sudden shifts in mood, or changes in mental health allowing a person to seek help if needed [2]. Finally, through a web service, public users could also collect demographics about the emotional state of a populace. Not only that, but medical organizations can also infer correlations between geographical conditions, context, and psychological wellbeing of individuals in that region.

Emotion recognition on various devices typically relies heavily on user input gathered in an intrusive manner [3] such as filling in surveys and/or questionnaires, or by using language processing [4] to determine the user's mood. Filling out forms is cumbersome and, for example, not likely to happen when someone is angry. Similarly, using natural language processing for emotion detection, especially on a phone, is difficult. For example, if someone were to type "lol" or "rofl" etc., the natural language processor, unless configured to recognize these short hands, would infer, falsely, that the user made a spelling mistake. Moreover, if the nature of language is taken into account and the way in which people develop words and short hand notations as new technology comes around (Google is not an English word, but is now used as a verb "let me Google that"), it becomes very difficult to design a system that can consistently detect a user's emotional state based on language alone [4].

This paper proposes to recognize the emotional state of a user by exploiting the various built-in sensors in a mobile phone. This is achieved by creating a soft-keyboard that uses sensor data to eventually determine a user's current emotion. This soft-keyboard replaces the default mobile phone keyboard and can be used with any application. The soft-keyboard connects to web-service that provides personalized statistics reflecting the emotional state of a user through time. Others can access the web service to view the average emotional profiles of populations across geographical locations.

The rest of the paper is organized as follows. The next section describes previous work in detecting emotions using mobile phones. Section III describes the design of the system including an evaluation of the machine learning algorithms used. Section IV shows the system architecture and implementation. Section V presents the conclusion and future work.

2. BACKGROUND

This section provides a brief overview of previous mobile phone applications that recognize user emotions. The use of machine learning for emotion recognition is also discussed.

2.1 Detecting Emotions Using Mobile Phones

Shivhari and Saritha [5] proposed key spotting method to classify the user's emotional state based on keywords found in the user's text input. The algorithm uses a six-step process that consist of 1) Capturing User's text input, 2) Tokenizing text, 3) Identifying keywords, 4) Analyzing keywords and weighing them on a preset scale quantifying the emotion, and 5) Adding the weights to create a final classification. There are two primary limitations with this method of emotion classification. The first limitation is that this method does not account for the context in which the words occur, but merely checks for the occurrence of specific keywords. The second limitation is that the algorithm does not consider user's word choice patterns as part of the classification process. Not considering word choice pattern leads to the output being inaccurate for a wide range of users [6].

EmotionSense [7] is a stand-alone application that works by first asking users to sign up to their web service through an email account. This is done to allow data gathering for later access by the user. After sign-up, the users are taken through a brief survey that asks them questions about their emotions followed by a question that asks users to select the intensity of their current emotions on a graph. For example, the user enters the intensity of moods like "calm" or "anxious." Based on manual input, the application plots the user's mood (positive vs. negative and sleepy vs. alert, for example) on a grid. In addition, the application uses the built-in sensors like the accelerometer and the GPS to determine if the user is active or not. Level of social interaction is measured by the amount of social media used. Every week, the app unlocks a new method of detecting the user's emotion. For example, in the second week, it unlocks detection using location, then SMS patterns, and so on. Every day, the app asks the user how he feels and adds their emotion to the output grid. This is done to allow the application to develop a baseline against which it can determine the user's emotional state based on phone usage information. The user is able to check his/her statistics at any time. It should be mentioned that moods are self-reported.

T2 Mood Tracker [8] is a stand-alone application that acts like a mood diary by frequently asking the user to rate how he/she feels. This is done through the use of sliders; one for each emotion. The app then plots the emotional data over time. The application allows the user to generate reports on dimensions like anxiety, depression etc. Unlike *EmotionSense*, this application does not perform a computational analysis of the user's device usage parameters. The app only determines the user's state from the data he or she provided manually.

2.2 Using Machine Learning for Emotion Recognition

Many machine learning algorithms attempt to automate text categorization [9]. For emotion recognition, the algorithm needs to classify a user's emotional state (e.g., angry) based on the provided user input (e.g., text being typed, sensor data etc.). The primary advantage of the machine learning approach is its ability to tailor the classification based on an individual user's behavior. Supervised machine learning algorithms are used to solve this class of problems. Algorithms in this category initially require data input to be labelled with the desired output. After the initial training period, the algorithm can begin to classify new input based on the pre-classified data that was originally provided. This paper considers multiple learning algorithms; Naïve Bayes, Support Vector Machines, J48 and Regression. Each learning algorithm is briefly described next.

Naïve Bayesian is a statistical technique for building classifiers. Every instance is represented as vector of finite set of attributes. All algorithms in the Naïve Bayes family share a common principle: all attributes are independent of each other; a value of a particular attribute contributes independently to the probability for a certain classification [10]. Despite their basic design and oversimplified assumptions, Naïve Bayesian classifiers are effective in numerous real and complex situations. These classifiers typically require a small amount of training data to estimate parameters. Even though Naive-Bayes approach does not always reflect the underlying model, the models thus constructed are still considered to be suitable. Bayesian classifiers have been used to detect emotions using facial expressions [11][12].

Another appropriate machine learning method is Support Vector Machines (SVMs) [13]. SVMs classify data by creating a set of support vectors, through a process of risk minimization. Support Vectors (SVs) are part of the training set that outline a hyperplane in feature space. This N-dimensional hyperplane (N is the number of features/parameters) defines borders/margins between the different possible classes [13]. Classification is based on determining which side of the hyperplane the test entry belongs to. By minimizing the structural risk the average error rate of classification is reduced. These borders are defined as the shortest possible distances to the nearest positive (as in true) on negative (as in false) point. The algorithm tries to separate hyperplane with the biggest margin/boarders. Many libraries for implementing SVMs are available. SVMs have also been used for emotional recognition using facial expressions and text analysis [13].

J48 decision tree method is a derivative of the C4.5 algorithm [14]. In J48 the training sample consists of a multi-dimensional vector, where every entry in the vector corresponds to an attribute/feature. At each node, the algorithm selects an attribute that splits the data set into different subsets most effectively. The attribute with the highest normalized information gain is selected to make the split decision [14]. J48 is more suitable when training relatively large amount of data. This algorithm can handle default data and noise, and has a high classification accuracy in some situations. In addition, this algorithm is also suitable to handle chaotic and complex data. J48 has been used for emotion recognition through facial expression and psychological signals [15].

The instance based learning or *k*-Nearest Neighbors algorithm is a rote learning method used for classification. Nearest neighbor algorithm are considered Lazy learners, where all calculations are deferred until a classification/prediction is required. The instances of the training set are called “knowledge”. The algorithm searches the training set for “knowledge” instance(s) that is most similar to the new instance that needs to be classified [16]. The similarity is determined by calculating the distance, which, for example, can be Euclidean distance or Manhattan distance. Classification is done based on the majority vote of the test instance’s *K* neighbors. The value of *K* should be set to a positive integer, typically small. If *K* is selected to be one, then the test instance is assigned to the class of the single nearest neighbor. The choice of *K* is very critical, a relatively small value entails that the effects of noise on the classification are more severe. On the other hand, a large value will make computations more complex and nullifies the simplicity of the algorithm. A common approach is to select *K* to be the square root of the number of instances within a training set and increment or decrement its values based on trial runs on the training set until accuracy is maximized [16].

Multi-response linear regression [17] is a meta-classifier. This technique builds separate linear regressions equations for each class where the output is set to one for all instances belonging to the class and to zero for others. The resulting regression line creates a threshold for predicting a class. All instances that belong to that class will have a value larger than the threshold and thus correspond to 1. Whereas all instances not belonging to that certain class will fail to exceed the threshold value and be assigned to binary 0. Since there are *N* regression equations for a problem with *N* different classes, if an instance exceeds the threshold for multiple classes at the same time, the instance is assigned to the class with the largest output, where the value of the threshold is largest in comparison to the other classes.

One potential problem with using users’ emotional responses for machine learning is that some emotional states may be rare. For example, it may be the case that the user is not angry much of the time. This makes the ‘angry’ state a rare event leading to un-balanced data. Rare events are those whose occurrence rate ranges from 0% to around 5% depending on the situation [18]. Classification of rare events is a common problem in many domains as the scarcity of a subset of the classes leads to an unbalanced dataset. An unbalanced data set will likely reduce the correctness of the classification. Initial data collection indicated for emotion detection using the mobile phone indicated that it was highly likely that a user will present one or two emotions for the majority of the time. Therefore, the rest of the emotions are under-represented. It seems that users tend to have a ‘default’ emotional state. This state varies between users; one user’s ‘default’ mental state was ‘neutral’ while another’s was ‘angry’. The ‘default’ mental state tends to occupy the datasets that are recorded (70% of the file on average); this will cause the data to be unbalanced in favor of one emotion over the others. The other reason for unbalanced data is that emotions are complex and for simplicity, this approach assumes that they are discrete. However, emotions are continuous and a user might experience an emotion that lies on the spectrum but does not fit into one of the emotions that are captured. This leads to one or more emotion being left behind and rarely recorded.

To handle unbalanced data, Synthetic Minority Over-Sampling Technique (SMOTE) [18] can be used. SMOTE generates instances/entries of the minority classes (under presented classes) by operating in feature space rather than the data space. By synthetically generating more instances of the minority class, the classifiers are able to extend their regions for the minority class. SMOTE uses the nearest neighbor computations for the minority classes such as Euclidean distance and Value Distance Metric. SMOTE works by using a majority vote between the feature vector under consideration and its k nearest neighbors for the nominal feature value, in the case of a tie, choose at random, and finally assign that value to the new synthetic minority class sample.

3. Emotion Detection Approach

The primary idea behind this approach is to collect sensor data while a user is typing on the keyboard. As a user types, he or she are prompted to indicate their current emotional state. In doing so, sensor data from the phone is tagged with the current emotional state of a particular user. Once enough data is collected, machine learning techniques are used to build classifiers that can predict the user’s current emotional states based on their current typing behavior. The approach proposed in this paper differs from previous work. First, the proposed system uses a soft-keyboard, and hence can be used with any application. This has the advantage of applying emotion recognition within the context of any application that uses the keyboard. Secondly, the proposed system is data-centric and automatically collects users’ data as they type on the keyboard in any application. Rather than always relying on the user to self-report their emotion, once enough data is collected, the application predicts emotions intelligently based solely on the sensor data from the mobile phone.

Figure 1 shows how the approach works. A user first installs the keyboard, and then activates it. After this step, a user can use this keyboard for any application. However, every time the keyboard is used, sensor and typing data is collected from the user. Then a machine learning algorithm is used to construct an emotion classifier based on captured data. In this training phase, the user is asked to indicate their emotion while they type. This recorded emotion is used as a tag, and the tagged data is used as an input train the classifier. After the training stage is over, the classifier takes the current typing behavior of the user and predicts their emotional state. Optionally, the emotional state data can be uploaded to a webserver for public usage.

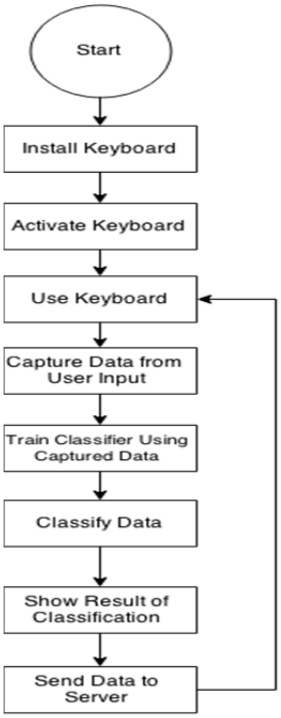

Start
Install Keyboard
Activate Keyboard
Use Keyboard
Capture Data from User Input
Train Classifier Using Captured Data
Classify Data
Show Result of Classification
Send Data to Server

Figure 1. The Emotion Detection Approach

The primary input to the machine learning algorithm is a set of feature vectors derived from typing behavior of the user. Each feature vector contains features representing a uniform segment of behavior. A feature vector consists of average acceleration, average time delay between typed letters, number of backspaces, and the associated user emotion. The first two components are calculated from mobile phone's sensor data, while the number of backspaces is recorded from the keyboard.

Using feature vectors as input, the following machine learning algorithms were evaluated to find the best classification method.

- Naïve Bayes – Estimates the probability of an entry being of a certain class based on previous entries.
- J48 Decision Tree – Creates a C4.8 decision tree that splits the data into different subsets.
- Lazy IBK – A nearest neighbor approach, where the distance between two feature vectors is calculated and a class is assigned based on the nearest neighbor.
- Multi-response linear regression – Classification possibilities are converted into binary and a regression model is created for each possible class.
- SVM – Creates a hyper-plane separating the various emotions.

Implementation of each algorithm in the Weka toolkit [19] was used to evaluate the alternative machine learning algorithms. The test set used contained 307 feature vectors, 109 were tagged as 'Neutral', 66 as 'Angry', 84 as 'Happy', and 48 as 'Sad'. The test set was collected over a period of one month from three volunteer users. Ten folds cross-validation was used. Cross validation splits the set into 10 parts. Every iteration uses 9 slices for training and the last slice as a test set. This is repeated until every slice is used a test set. Finally, each algorithm iterates the set for the eleventh time using the full set for testing.

In order to evaluate the performance of the algorithms, Precision-Recall and Receiver Operating Characteristic (ROC) curves for the various states used. As seen in Figure 2, multi-response linear regression and J48 performed well overall as compared with other algorithms. SVM seems have performed the worst among all the algorithms tested.

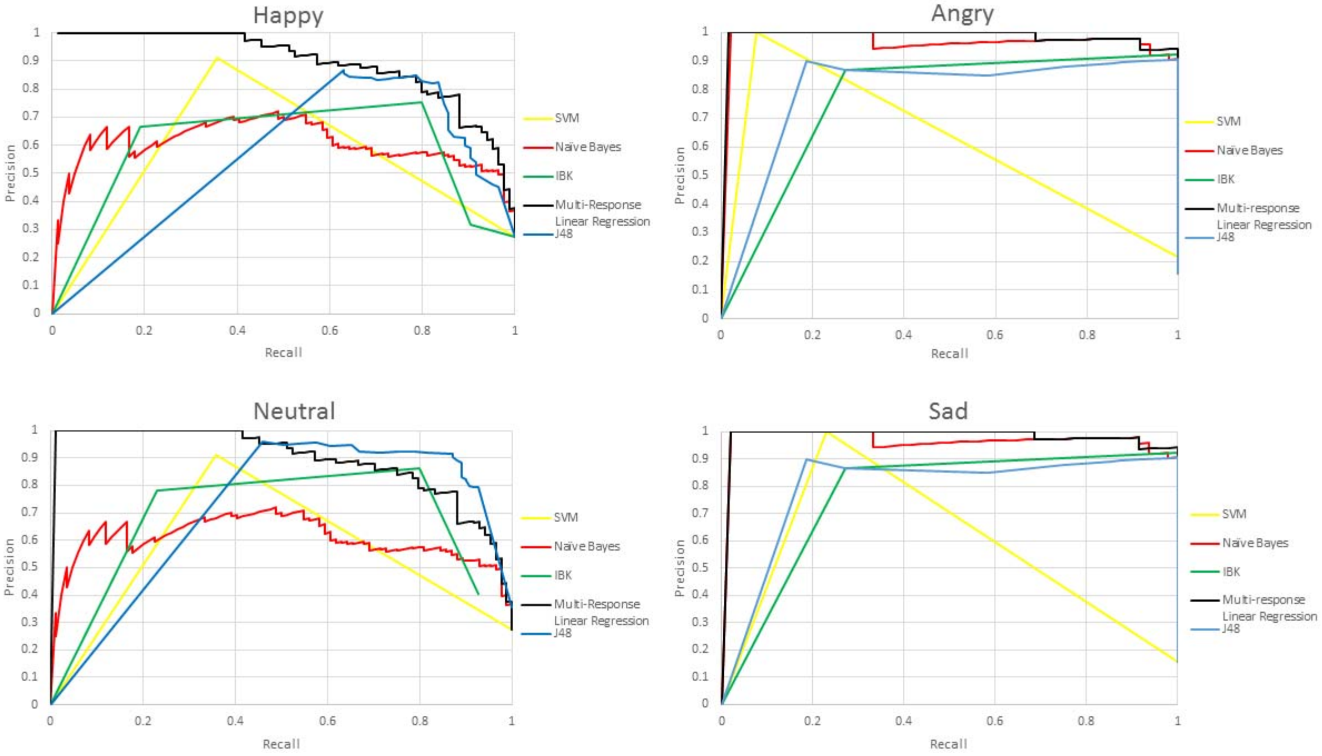


Figure 2. Precision-Recall Curves for All Four States

The ROC curve compares the true positive rate or correctly classified instances against the false positive rate or incorrectly classified instances. A perfect classifier has an upside down ‘L’ shape while the worst classifiers have a diagonal ROC curve. Figure 3 shows ROC curves for every emotional state as opposed to the others. As the Figure shows, SVM is the worst performing classifier because its ROC curve is diagonal. The two best performing classifiers are Multi-response linear regression and J48. It is important to note that the results use SMOTE. Therefore, the actual performance of the algorithm may be slightly different than the evaluation. The fact that SVM performed the worst and J48 seems to suggest that it is not possible to construct linear planes through the data space. Rather, specific hyper cubes within the space represent the various emotional states.

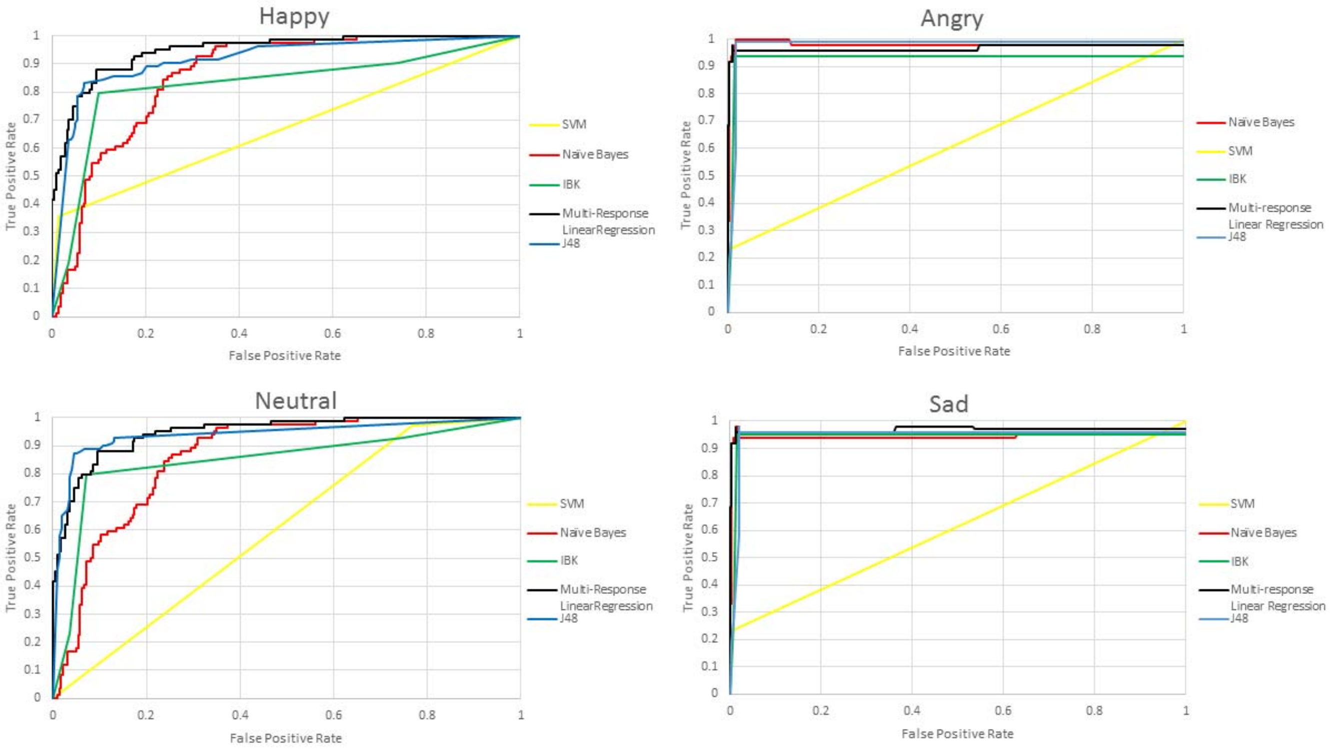


Figure 3. ROC Curves for All Four States

Based on ROC and Precision-Recall curves, it was concluded that both J48 and Multi-response linear regression performed well. The confusion matrixes for SVM, J48 and Multi-response linear regression are shown in Tables 1-3.

Table 1. The Confusion Matrix for Multi-response linear regression.

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.81	0.067	0.819	0.81	0.814	0.745	Happy
	0.881	0.071	0.873	0.881	0.877	0.808	Neutral
	0.985	0	1	0.985	0.992	0.99	Angry
	0.958	0.012	0.939	0.958	0.948	0.939	Sad
Avg.	0.896	0.045	0.896	0.896	0.896	0.851	

Table 2. The Confusion Matrix for J48.

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.821	0.067	0.821	0.821	0.821	0.918	Happy
	0.872	0.051	0.905	0.872	0.888	0.94	Neutral
	1	0	1	1	1	1	Angry
	0.979	0.019	0.904	0.979	0.94	0.987	Sad
Avg.	0.902	0.039	0.902	0.902	0.902	0.954	

Table 3. The Confusion Matrix for SVM.

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.357	0.013	0.909	0.357	0.513	0.672	Happy
	0.972	0.768	0.411	0.972	0.578	0.602	Neutral
	0.076	0	1	0.076	0.141	0.538	Angry
	0.229	0	1	0.229	0.373	0.615	Sad
Avg.	0.495	0.276	0.766	0.495	0.434	0.609	

From Tables 1 and 2 since Average F-Measure and ROC Area for J48 are better than Multi-response linear regression, J48 was selected to be the better performing algorithm. Table 3 shows clearly that the SVM approach does not work well since Precision, Recall, and F-Measure are all below 60%.

As Multi-response linear regression and J48 showed the most promise, the models generated by these algorithms will be further examined. Equations (1) - (4) show the class assignment process the Multi-response linear regression algorithm follows during the classification phase.

$$Emotion\ (Happy) = -0.0001 * letters - \ 0.003 * \text{Acceleration} \ + \ 0.0071 \tag{1}$$
$$Emotion\ (Neutral) = 0.0038 * letters \ + \ 0.0117 * timebp \ - \ 0.0901 * Acceleration \ - \ 1.4757 \tag{2}$$
$$Emotion\ (Angry) = 0.0017 * letters \ - \ 0.0002 * timebp \ - \ 0.0901 * Acceleration \ + \ 1.4281 \tag{3}$$
$$Emotion\ (Emotion) = -0.0002 * letters \ - \ 0.0002 * timebp \ - \ 0.0696 * Acceleration \ + \ 0.6354 \tag{4}$$

Figure 4 below shows the decision tree generated by J48. It is important to note that the threshold values (e.g. Acc <= 5.4366) are automatically generated by the algorithm during the training phase. Moreover, the tree shows that ‘Angry’ emotion is easily detected by using the Accelerometer data only as shown in the right branch of the tree. However, a complex set of decisions is required to differentiate ‘Neutral’ from the ‘Happy’ state as shown by the left sub-tree.

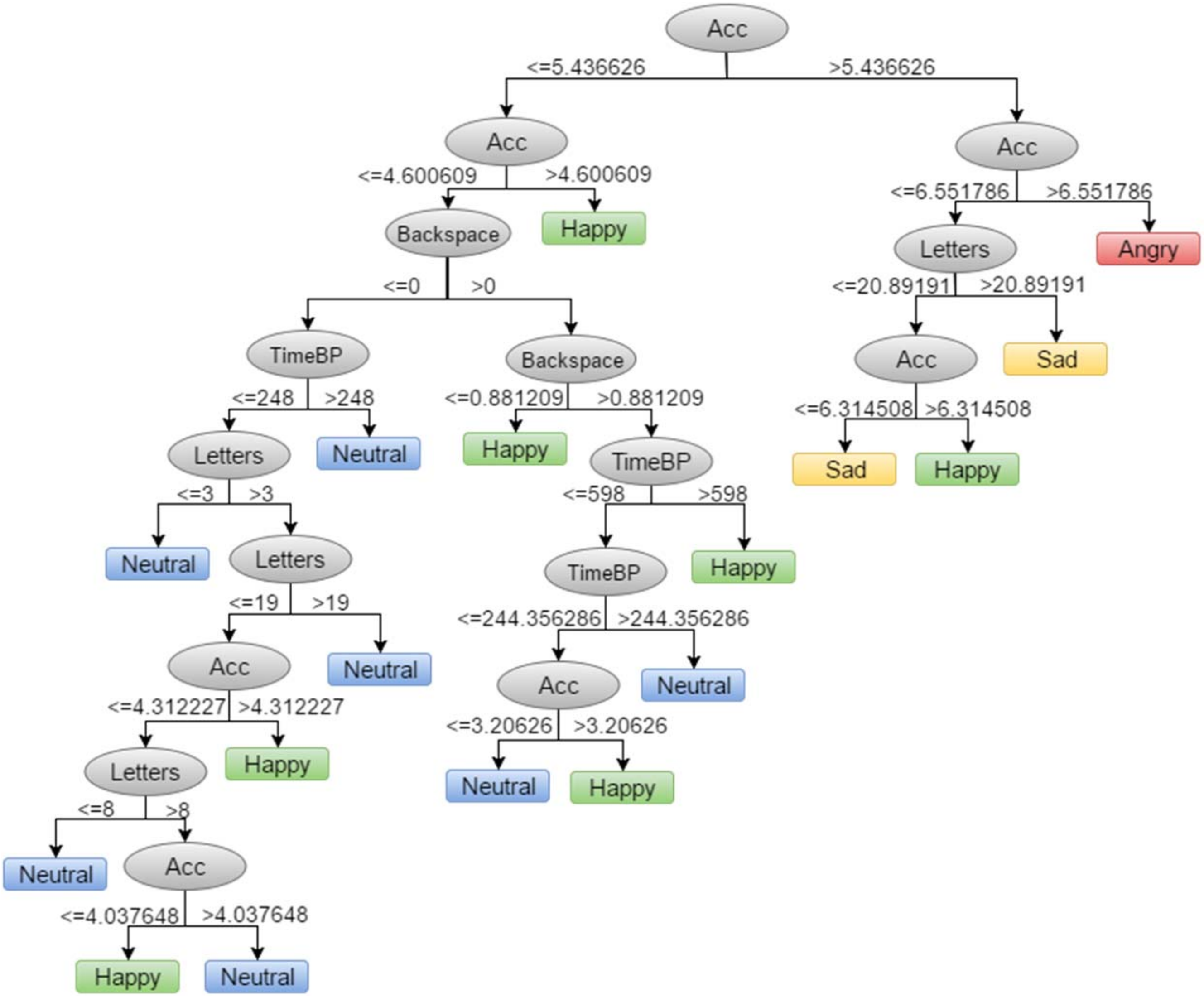


Figure 4. Decision Tree Generated by J48. “TimeBP” indicates Average Time between Key Strokes in milliseconds.

In summary, the emotion detection approach uses mobile phone's sensor data and user's typing behavior to train a machine learning algorithm to predict user's emotions. Experiments indicate that Multi-response linear regression is the best machine learning algorithm for this approach. Thus, will be used for the classification.

4. System Architecture and Implementation

Figure 5 shows the basic system architecture. As shown, there are two types of users: an *Application* User and a *Web* User. An Application User interacts with the mobile phone while the Web User uses the browser to view public trends.

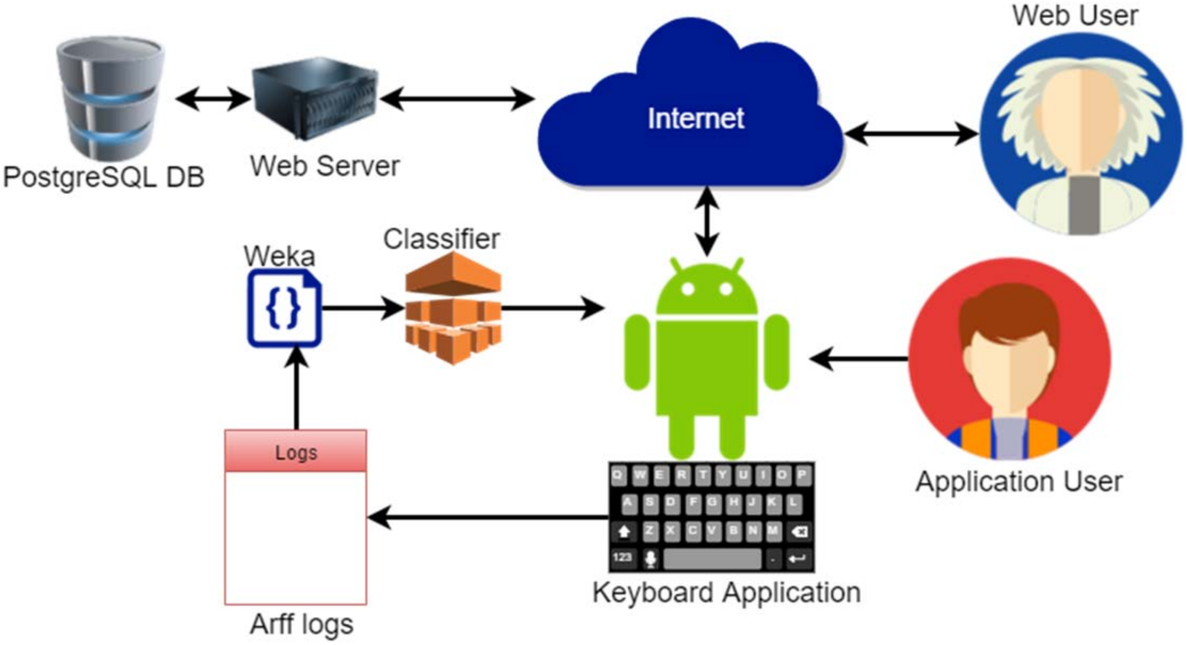


Figure 5. Basic System Architecture.

Moreover, the core of the system is the soft keyboard application running on the Android phone. This Keyboard Application is responsible for collecting sensor data and applying machine learning to predict the current emotion of the learner. Multi-response linear regression classifier from Weka's library was modified to run on the Android-based devices. The ported library supports any mobile phone after Android V2.0 (Eclair). The Keyboard Application stores feature vectors in the form of the log file in WEKA's ARFF format. This file is used as input to train the classifier. As Figure 5 shows, the soft keyboard also publishes user's emotional state into a web application. This is done through RESTful API. The Web Server is implemented using Python and generates geographical and other charts that depict the average emotional state of countries with registered users. Personalized charts depicting an individual's emotional state can only be accessed by the relevant user. The web-application uses the PostgreSQL DB to store all the data obtained from users.

The Keyboard Application and the Web Server are described in more detail below.

1. Keyboard Application

The Keyboard Application was developed by modifying the Android Open Source Project (AOSP) keyboard [20] to allow it to capture data while the user is typing. The response time of the keyboard was also improved to allow for a smoother user experience. The key board captures the time between key presses in milliseconds and the number of backspaces as a measure of mistakes made in input by the user. This is all captured periodically during a period of 5 seconds, wherein each period is denoted as a segment.

In the first phase, the algorithm requires the user to enter his current emotion in order to construct the training set which consists of pre-classified data. This is done through utilizing the emotion bar on top of the keyboard as seen in Figure 6. It has been empirically determined that the algorithm requires at least 150 segments to start predicting the user's emotional state within a reasonable margin of error.

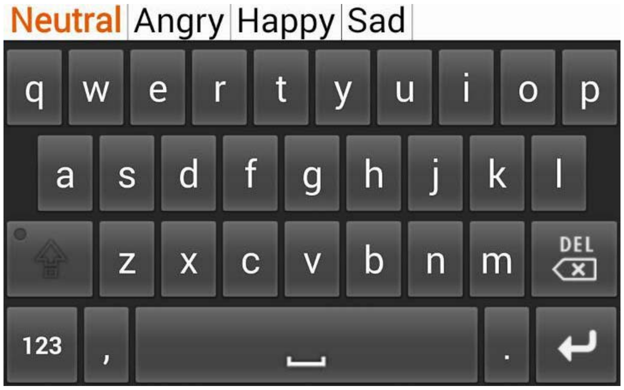


Figure 6. Emotion Bar Options.

After the training phase is over, the keyboard data is used to predict the current emotion of the user using the Multi-response linear regression algorithm. As Figure 7 shows, when used, the keyboard initiates a session that spans for five seconds. Within a session, the keyboard logs user typing behavior, acceleration in three dimensions and the emotion recorded from the candidate bar. This data is stored in a log file which is then processed to create a vector in the feature space which is appended to the *Record* file as a Record. At the end of every session, the algorithm is re-trained based on all previous entries in the *Record* file and the latest entry is passed through the regression algorithm to be classified. The classified entry is displayed on the candidate bar of the keyboard and is added to the emotion attribute of the last entry in the file.

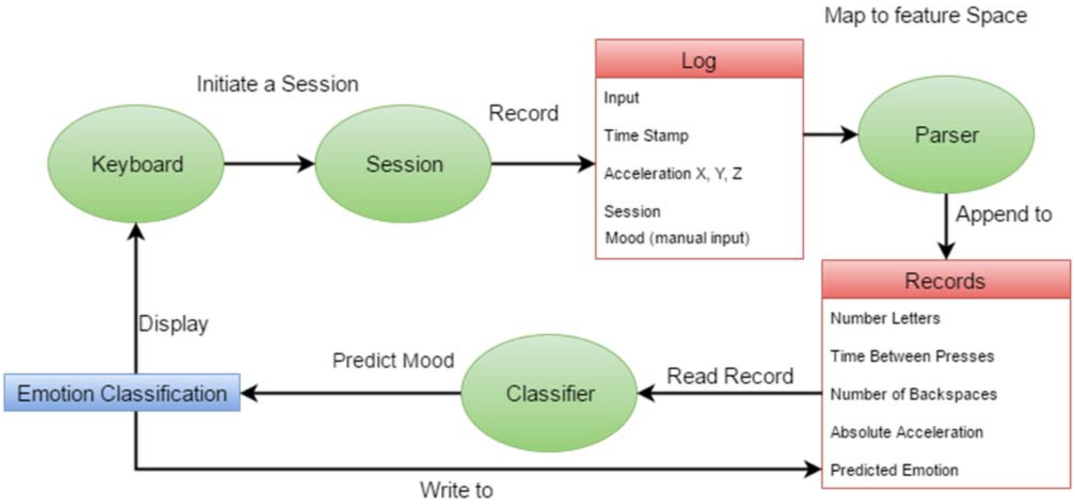
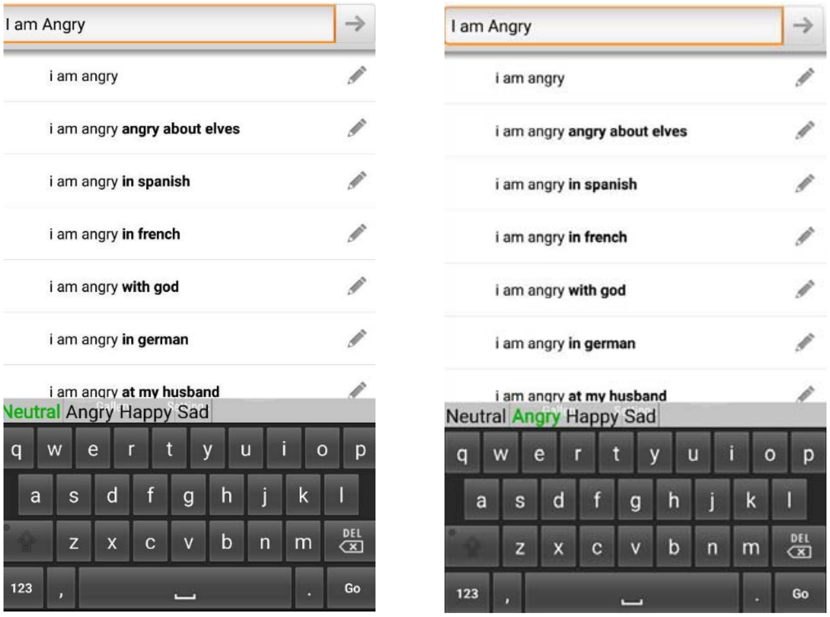


Figure 7. Data Capture and Prediction.

Figure 8 shows how depending on user's typing behavior, the current emotive state is changed from 'Neutral' to 'Angry'.


I am Angry

i am angry

i am angry **angry about elves**

i am angry **in spanish**

i am angry **in french**

i am angry **with god**

i am angry **in german**

i am angry **at my husband**

I am Angry

i am angry

i am angry **angry about elves**

i am angry **in spanish**

i am angry **in french**

i am angry **with god**

i am angry **in german**

i am angry **at my husband**

Figure 8. User's predicted state changing based on their input.

Finally, when the user first installs the keyboard, they are greeted with a log-in screen which allows them to log-in or to create a new account. Since the keyboard will not keep track of the user's emotions if the user is not logged in, the log in screen keeps popping up until the user logs in. When the user logs-in, the keyboard receives a token which is then stored on the phone for later use. After logging in, at certain time intervals, the current emotion of the user is sent to the Web Server along with a timestamp and the geolocation of the user which is found based on the user's service provider. If the user does not have a service provider, he or she is given the option to manually change their location from the keyboard's settings menu. This eliminates the need for the use of GPS and location services. The emotion, timestamp, and country are then sent to the server via a HTTP POST request. The token received earlier when the user logged-in is placed as a header for the POST request, with the emotion, timestamp, and country placed in the body of the request.

2. Web Server Implementation

The Web Server is implemented in Python using a web framework called Django [21]. Below is a breakdown of the elements in the backend and the way they were implemented.

At the core of the WebServer is the PostgreSQL database that stores the information about the users registered to the website, and the information about their emotional states. Django comes with an Object-Relational Mapping (ORM) out of the box. This allows for the creation of classes that directly map into SQL tables. Furthermore, the ORM can generate query sets using class methods instead of relying on SQL queries which may be vulnerable to SQL injection. Figure 9 shows the ER diagram which contains the '*auth_user*' table used for storing user information and the '*emotiondata*' table used to store the data sent by the user's device.

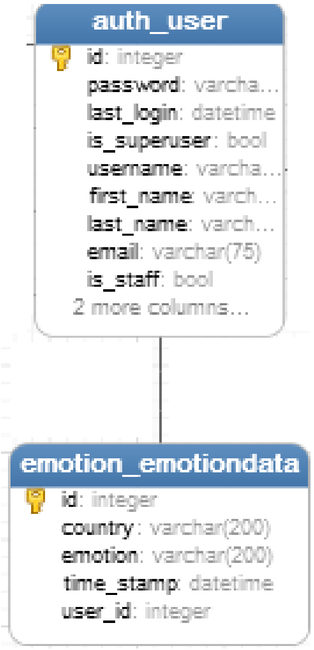
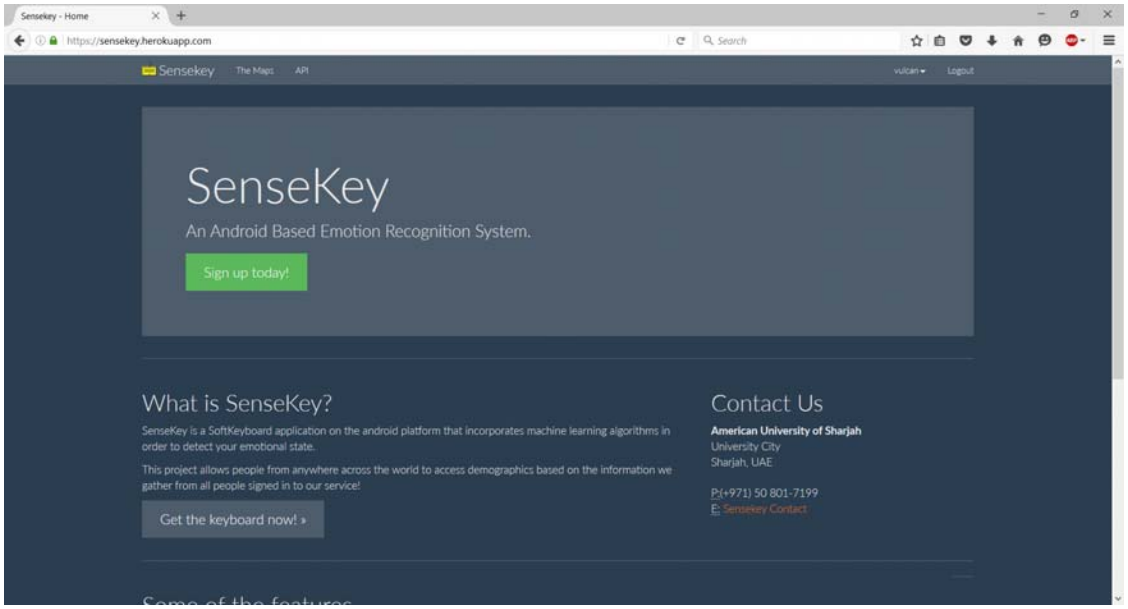


Figure 9. ER diagram for the user and emotion data.

The browser-based User Interface (UI) for the Web Server was designed by using the bootstrap3 framework [22] which allows for rapid integration of common, responsive, designs. The website's navigation is consistent across the entire website in order to maintain a consistent experience. Furthermore, Django's messaging middleware was used to allow for responsive messages that inform the user of what is going on as he or she interacts with the Web Server. Another advantage to using bootstrap is that it creates responsive views that scale to a user's viewport size. In other words, if the user was using a mobile phone to access the website it would adapt to suit the smaller screen size. The front page of the Web site can be seen in Figure 10. The HighCharts library [23] was used to create personalized reports for individual users. Figure 11 shows a personalized report on a mobile phone.



SenseKey

An Android Based Emotion Recognition System.

What is SenseKey?

SenseKey is a SoftKeyboard application on the android platform that incorporates machine learning algorithms in order to detect your emotional state.

This project allows people from anywhere across the world to access demographics based on the information we gather from all people signed in to our service!

Contact Us

American University of Sharjah
University City
Sharjah, UAE

P:(+971) 50 801-7199
E: Sensekey Contact

Some of the features

Figure 10. Homepage Screenshot.

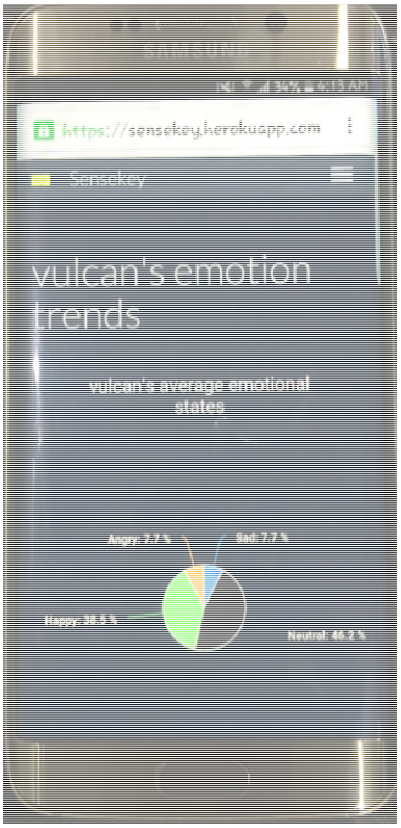


Figure 11. Showing personalized reports on a mobile phone.

To provide maps that depict the emotional states across regions, Google's GeoCharts [24] was used to generate a choropleth that illustrates the modal emotional state across the region. This choropleth is updated with every new data point posted to the Web Server. A sample map can be seen in Figure 12 where each color shows the most prevalent emotive state.

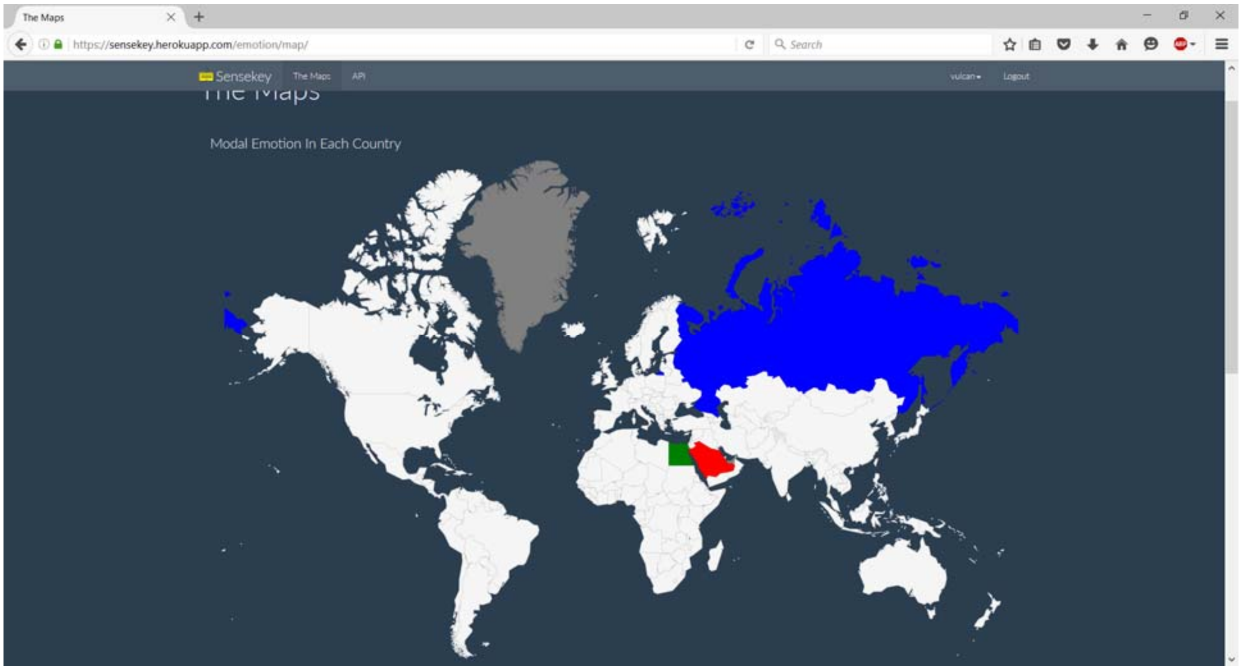

The Maps
https://sensekey.herokuapp.com/emotion/map/
Search
Sensekey
The Maps
API
vulcan
Logout
The Maps
Modal Emotion In Each Country

Figure 12. Average Emotion States of each country.

5. Conclusions and Future Work

This paper presented a machine learning approach for emotion recognition using a mobile phone soft keyboard. The keyboard records the user's typing behavior that includes texting speed and time between presses, and shaking as measured through the built-in accelerometer. The keyboard dynamically uses the Multi-response linear regression machine learning algorithm in order to classify the user's current mood. The system also sends anonymized user data to a server that can be publicly accessed to view demographic information. The demographic data could be used by researchers in various fields and disciplines. The system demonstrates that it is possible to enable emotion recognition on mobile phones using built-in sensors. The system also does so in an application independent manner where any mobile application using a keyboard for input can use the service.

The current work has limitations and can be improved in multiple ways. First, classification accuracy, though high, can potentially be improved by incorporating additional attributes. For example, location (home, work, etc.), time, intensity of finger strokes, usage of strong language, facial expression, ambient temperature, weather data and discomfort index can be considered. These additional parameters can potentially help improve the accuracy rate and allow one to add more emotional states to the list of emotions the application can detect. Secondly, the keyboard layout can be made more appealing to the user as well. Finally, on the Web Server side, third-party login services can be incorporated to allow users to quickly sign up to the service.

6. References

[1] K. Button, G. Lewis, and M. Munafò, "Understanding emotion: Lessons from anxiety," *Behavioral and Brain Sciences*, 35(3), P. 145, June 2012. DOI: 10.1017/S0140525X11001464.

[2] M. Munezero, C. Montero, E. Sutinen, "Are They Different? Affect, Feeling, Emotion, Sentiment, and Opinion Detection in Text," *IEEE Transactions on Affective Computing*, 5(2), pp. 101–111, April 2014. ISSN: 1949-3045. DOI: 10.1109/TAFFC.2014.2317187.

[3] A. Agrawal and A. An. "Unsupervised Emotion Detection from Text Using Semantic and Syntactic Relations," in Proc. of the *IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology*, vol. 1, pp. 346–353, December 2012. DOI: 10.1109/WI-IAT.2012.170.

[4] Z. Su, R. Yan, L. Zhang, S. Xu, S. Bao, D. Han, and Y. Yu, "Mining Social Emotions from Affective Text," *IEEE Transactions on Knowledge and Data Engineering*, 24(9), pp. 1658–1670, September 2012. DOI: 10.1109/tkde.2011.188.

[5] S. Shivhare and S. Saritha, "Emotion Detection from Text Documents," *International Journal of Data Mining & Knowledge Management Process*, 4(6), pp. 51–57, November 2014. DOI: 10.5121/ijdkp.2014.4605.

[6] Cheng-Yu Lu, W. Hsu, and H. Peng, "Emotion Sensing for Internet Chatting: A Web Mining Approach for Affective Categorization of Events," in Proc. of the *IEEE International Conference on Computational Science and Engineering* (CSE), December 2010. DOI: 10.1109/cse.2010.44.

[7] G. Sandstrom, J. Rentfrow, and C. Mascolo, "EmotionSense," *University of Cambridge and University of Essex*. URL: *http://emotionsense.org/*

[8] The National Center for Telehealth & Technology. T2 Mood Tracker. URL: *http://t2health.dcoe.mil/*

[9] F. Sebastiani. "Machine learning in automated text categorization," *ACM computing surveys*, 34(1), pp. 1–47, March 2002. DOI: 10.1145/505282.505283

[10] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," *AAAI workshop on learning for text categorization*, pp. 41–48, 1998.

[11] I. Cohen, N. Sebe, Y. Sun, M. Lew, and T. Huang, "Evaluation of expression recognition techniques," *International Conference on Image and Video Retrieval* (CIVR), Springer, LNCS 2728, pp. 184–195, 2003.

[12] N. Seba, M. Lew, I. Cohen, A. Garg, and T. Huang, "Emotion Recognition Using a Cauchy Naive Bayes Classifier," in Proc. of the *International Conference on Pattern Recognition*, pp. 17–20, August 2002.

[13] M. Dumas, "Emotional expression recognition using support vector machines," in Proc. of the *International Conference on Multimodal Interfaces*, 2001.

[14] S. Ali, S. Zehra, and A. Arif, "Performance Evaluation of Learning Classifiers for Speech Emotions Corpus using Combinations of Prosodic Features," *International Journal of Computer Applications*, 76(2), 2013. DOI: 10.5120/13221-0634.

[15] C. Nie, J. Wang, F. He, and R. Sato, "Application of J48 Decision Tree Classifier in Emotion Recognition Based on Chaos Characteristics," in Proc. of the *International Conference on Automation, Mechanical Control and Computational Engineering*, April 2015. DOI: 10.2991/amcce-15.2015.330.

[16] R. Duda, P. Hart, and D. Stork, "Pattern classification," *John Wiley & Sons*, 2nd edition, November 2012. ISBN: 978-1-118-58600-6

[17] I. Witten and E. Frank, "Data Mining: Practical machine learning tools and techniques," *Morgan Kaufmann*, 3rd edition, 2011.

[18] N. Chawla, A. Lazarevic, L. Hall, and K. Bowyer, "SMOTEBoost: Improving prediction of the minority class in boosting," in Proc. of the *European Conference on Principles of Data Mining and Knowledge Discovery*, LNAI 2838, Springer, pp. 107-119, 2003.

[19] Weka 3 - Data Mining with Open Source Machine Learning Software in Java. URL: *http://www.cs.waikato.ac.nz/ml/weka/* (visited on Dec. 4, 2016).

[20] Android Open Source Project. URL: *http://source.android.com/* (visited on Dec. 4, 2016).

[21] Django, The web framework for perfectionists with deadlines. URL: *http://www.djangoproject.com/* (visited on Dec. 4, 2016).

[22] Bootstrap. URL: *http://getbootstrap.com/* (visited on Dec. 4, 2016).

[23] HighCharts. URL: *http://www.highcharts.com*/ (visited on Dec. 4, 2016).

[24] Visualization: GeoChart. URL: *http://developers.google.com/chart/interactive/docs/gallery/geochart* (visited on Dec. 4, 2016)

Imran Zualkernan holds a B.S. (High Distinction) and a Ph.D. from the University of Minnesota in Minneapolis. His research is in advanced learning technologies and Internet of Things. He has published over 120 research papers in international conferences, workshops and journals. He has served as a CEO and CTO and has designed and deployed advanced commercial robotics and Internet-based systems.

Fadi Aloul received the B.S. degree in electrical engineering (*summa cum laude*) from Lawrence Technological University, Southfield, MI, and the M.S. and Ph.D. degrees in computer science and engineering from the University of Michigan, Ann Arbor. He is currently a Professor of Computer Science and Engineering and the Director of the HP Institute at the American University of Sharjah, UAE.

Shams Eddeen Shapsough received his B.Sc. in Computer Engineering from the American University of Sharjah (AUS) in 2015. He is currently pursuing his M.Sc. degree in Computer Engineering at the same university.

Ahmed Awad completed his B.Sc. from the American University of Sharjah (AUS) in 2015, and his M.Sc. in Networked Computer Systems from University College London (UCL) in 2016. He currently lives in England where he is pursuing a Ph.D. in computer science at UCL.

Youssef El-Khorazaty born in 1994 in Jeddah, received his B.Sc. in Computer Science from the American University of Sharjah (AUS) in 2015. He currently lives in Cairo, Egypt where he works as Technical Support at Dell EMC.