

# Detect Individual KeyStrokes with Smartwatch Sensors

Anuj Kumar

College of Information and Computer Sciences

University of Massachusetts Amherst

anujkumar@umass.edu

## Abstract

*Wearable consumer electronics with a myriad of sensors are becoming increasingly popular. With the boost in number of smartwatch users leading this drive, the varied applications of a smartwatch are also on the rise. This project is aimed at a new such use of a smartwatch-like device: to detect keystrokes typed on a classic keyboard.*

## 1. Introduction

In the era of ubiquitous computing, it is becoming increasingly popular to rely on smart devices to better our standard of living. Smartwatches currently are undergoing the same boost in popularity that smartphones underwent about a decade ago. With more and more users opting for these devices, the expectations on the capabilities of these devices also increases. Recent advances have been made towards activity recognition [1] and fitness tracking [4] based on data collected off smartwatch sensors such as accelerometer and gyroscope.

On an orthogonal approach, other works have shown that smartwatches also offer a security threat to our privacy through leaked data of the sensors. Wang et. al. [2] and Maiti et. al. [3] show that it is possible for a hacker to reconstruct approximate typed words based off just the accelerometer sensors on the smartwatch worn on one hand by the user. These methods caution against unaudited access to smartwatch sensors. However, as long as there is awareness of such scenarios, these issues are not a major problem. For example, we type our passwords directly into laptops and mobiles without worrying about others eavesdropping. In other words, our passwords are kept secure by following security protocols and not by limiting the different uses of these computing devices themselves.

Extending this thought, the ability of a smartwatch to recognize typed keys can be a blessing in disguise. Accelerometers and gyroscopes are basic sensors that can be found in a myriad of devices (spectacles, phones, watches, etc). It is probable that in the near future, people might wear

another device on their other hand which has these sensors, hence rendering accessible the sensor data from both hands (one from watch, the other from the other band). Using these sensors to capture data from both the hands of a user, and to recover typed keystrokes from such data would be the next step. Earlier works [2] [3] relied on a pre-built model of the English dictionary, and thus were limited to correctly-typed English words and also information from only one watch. In real life, we often make mistakes and often use slangs and words from a different language during communicating. If these sensors could detect individual keystrokes, we could then build a model where we wouldn't need a keyboard to type and could do so simply by imitating the keyboard motion with our fingers.

## 2. Problem Statement

As part of the Neural Networks course, this project aims at detecting keystrokes from smartwatch data. In particular, we are interested in detecting individual keystrokes from accelerometer and gyroscope sensors of the two smartwatches worn on each hand of a user. A secondary goal could also be to predict, based on pre-built words of English dictionary or of user typing pattern behaviour analysis, the character that best fits the current scenario, and then use it to auto-correct typing mistakes by the user. This goal will probably be beyond the scope of this course project and will only be explored given time.

Typing datasets of users typing with smartwatches worn on both wrists aren't available on the internet. Hence, for this project, I am responsible for collecting my own data. My aim is to build a model that can correctly predict what character a particular user is typing. To this extent, taking data from a single user is deemed substantial for the project. I have collected data myself after having worn an ASUS ZenWatch 2 on my left wrist and a LG G Watch W100 on my right wrist. I used a self-modified version of the publicly available app Sensor Data Logger [5] to record data from these smartwatches on two smartphones. This data is collected at 50Hz for accelerometer, gyroscope and gravity sensors. Gravity sensors are kept to be used later to boost

performance. This app captures the timestamp, accurate to milliseconds, and the readings of the nine axes of sensors. For ground truth, I have used the publicly available Key Frequency Logger project [6], which records each keystroke with a timestamp accurate to microseconds.

Using the above scenario, I have captured typed data over 6 hours of typing wearing both smartwatches. I have then processed the data and cleaned it, so that I consider the keystrokes only belonging to a particular character set (a-z for example), and consider sensor readings only just before and immediately after these keystrokes. Considering only the alphabetical character set, the processed data labels comes out to be around 10,000 typed characters currently, as recorded by the Logger [6], and the data corresponds to these labels. The labels consist of timestamps and struck keystroke whereas the data consists of timestamps and sensor readings. I might add to this data set later. Majority of my time until now has gone into preparing the frameworks for data-collection [5][6] and then processing this data. The complete data set and the processing code files can be found at my repository for this project <https://github.com/anujkumar93/DetectKeystroke>.

### 3. Technical Approach

With the current dataset, I plan to classify keystrokes using the mechanisms suggested in [3] and [2]. This starts by considering each keystroke an independent activity and only looking at the sensor window just before and after this keystroke. Then, a feature space is built by first converting the time-series data to frequency domain using the Fast Fourier Transform, and then taking a range of statistical features on this data (such as the mean, std dev, skewedness, kurtosis, etc). This set of features is then trained upon using a (50-layer) deep neural network to predict the character label.

The current issue with this approach is that the papers aren't very clear on their methodology. Further my typing speed is about 2-3 characters per second. This translates to only about 20 sensor readings for one typed character. This is a very small window to work with. The Fourier transforms over such a window will not be accurate and the statistical features may not have a large significance. Although this approach hasn't been tried by me yet, I believe it might be more beneficial to start off by converting the complete signal into frequency domain and then chunking the data. Also, I plan to experiment with shallower networks and a smaller dataset (1000 characters) first before diving into deep networks for classification. Once I am ready with my feature space, and if I believe they make more sense as images, I would experiment with Convolutional NNs to check for accuracy boost.

Since I don't have a pre-built model of the English Dictionary, I also wish to predict these characters using RNNs.

This would work by taking the time series data at each step and passing it through a vanilla RNN architecture and computing the output at the timestamps of the keystrokes and then comparing with the ground truth. Training a RNN this way might be beneficial in the cases where two consecutive letters are prone to appear together elsewhere too. For example, the letter 'u' almost always succeeds the letter 'q' and the RNN should be able to predict this pattern.

As a final step, I would use the outputs of both the networks and combine them to get a keystroke classification model. This model would be evaluated by calculating its accuracy in predicting the correct character as compared to the ground truth.

### 4. Preliminary Evaluation

The affine NN model discussed above is currently being built, and has not reached its completion and tried on the dataset. The dataset took multiple iterations due to the buggy nature of the Sensor Data Logger app [5] to come out clean, at the end of which I have about 10k readings of the logged characters of the English alphabet and correspondingly about 17k readings for the sensor axes. While logging and analyzing data, it is perceptible to the naked eye that logging characters has a marked effect on sensors of the right hand than of the left hand, implying that the right hand moves more and covers a larger area as compared to the left hand for me. It is also deducible that the same character produces very similar curves in sensor readings and thus, the separation of signal into each individual character should be inherently possible.

### References

- [1] Ravi, N., Dandekar, N., Mysore, P. & Littman, M. L (2005) Activity Recognition from Accelerometer Data. *Innovative Applications of Artificial Intelligence*, pp. 1541–1546.
- [2] He Wang, Ted Tsung-Te Lai, & Romit Roy Choudhury (2015) MoLe: Motion Leaks through Smartwatch Sensors. *MobiCom 2015*
- [3] Anindya Maiti, Oscar Armbruster, Murtuza Jadliwala, & Jibo He (2016) Smartwatch-Based Keystroke Inference Attacks and Context-Aware Protection Mechanisms. *ASIA CCS 2016*
- [4] Li X, Dunn J, Salins D, Zhou G, Zhou W, Schssler-Fiorenza Rose SM, et al. (2017) Digital Health: Tracking Physiological and Activity Using Wearable Biosensors Reveals Useful Health-Related Information. *PLoS Biol*15(1)
- [5] *Sensor Data Logger* - open-source Android Wear app, available on GitHub at <https://github.com/Steppschuh/Sensor-Data-Logger>. Also available on Play Store for direct installation.
- [6] *Key Frequency Logger* - open-source Git Project, available at <https://github.com/bagder/keyfreq>