Using Integrated Smartwatch Sensing and a J48 Decision Tree for Better Teeth-Brushing Habits

Gavin Crigger
Computer Science, CLAS
University of Virginia
Charlottesville, Virginia
tzn4fx@virginia.edu*

Matthew Lucio
Computer Science, SEAS
University of Virginia
Charlottesville, Virginia
bgc9yp@virginia.edu*

Hunter Oakey
Systems Engineering, SEAS
University of Virginia
Charlottesville, Virginia
hho2em@virginia.edu*

Abstract—**With the rise of IoT devices and signal/sensing capabilities, the field of smart health applications is becoming exceedingly open for developers to enter. We propose a development pipeline for gathering data, training a classification model, and implementing feedback control on a smartwatch. This application is specifically geared towards encouraging healthy toothbrushing habits, using positive reinforcement and daily reminders (guided by an ML toothbrush habit detection algorithm). We found the mean X and Y acceleration to be the most influential features in 1-second intervals and developed a decision tree with an accuracy of 97.0126%, which was implemented on an Asus ZenWatch 2 Model W1501Q using cascading if-else statements. Some issues arose when extending this model onto a full application. Further, our work could have had better external validity: instead of a smartwatch, the sensor could be directly integrated into the toothbrush; if an activity is specific to hand dominance, then the signals and their sources used to detect them should reflect that.**

Keywords—**smartwatch, signal processing, classification, sequential feature selection, decision trees, machine learning**

I. INTRODUCTION

Smart device availability has increased tremendously within the past two decades, enabling unprecedented levels of independent research into how daily biometric signals and software can be incorporated into daily lives. We employ continuous signal processing, machine learning, and basic concepts of control into a smartwatch application. This uses accelerometer readings to identify when and for how long a user is brushing their teeth and nudges them to brush for at least 2 minutes. The aim of this project is to encourage healthy tooth brushing habits, developing a methodology for incorporating simple signals, like everyday hand motions, into useful feedback systems.

II. MATERIALS AND METHODS

We used an Asus ZenWatch 2 Model WI501Q smartwatch to record signals and execute our application. Signal recording was done both preliminarily (for training our classification model) and continuously (for monitoring user actions in accordance with our application). Our application was built and loaded into the device using Android Studio SDK API 23, and it was programmed in Kotlin. The exact smartwatch was number 23 from our CS 6762 course collection, and it was loaded with our android application for qualitative testing.

For representative training samples, we collected data for both the hand motion for brushing teeth and other ambient and active motions users may perform throughout the day (driving, walking, drinking water, etc.). After collection, we preprocessed and extracted features from our raw data files for analysis. To do this, we trimmed the first 0.5 seconds and the last partial second of each file to minimize the effect of starting or stopping during data collection, and to round its length to the nearest full second. For each file, we then normalized the timestamp to 0 seconds (by subtracting the first timestamp value from the entire column), and then aggregated the average and standard deviation of the accelerometer data, grouping by each second. After this, we added a column to label each row in each file as "BrushTeeth" and "NonBrushTeeth", based on how we tagged the activity with the smartwatch, and we stacked all of the files together. Lastly, we removed all columns that were not relevant to this data analysis task, including the timestamp. Our final data set included 672 seconds of toothbrush data and 3780 seconds of non-toothbrush data.

We employed sequential feature selection on these data to evaluate effective features while keeping our decision tree relatively simple (to avoid over-tuning our model). To do this, we retrained the model on sequentially added features until the next feature would add less than 0.5% accuracy. These features were X_acc_mean and Y_acc_mean, which correspond to the mean acceleration in the x direction and the mean acceleration in the y direction, respectively. The resulting decision tree was implemented into our app by replicating the decision splits, which we determined by using WEKA, with cascading if-else statements. Our app used this to classify each passing second as brushing or nonbrushing, accumulating these classifications over time to determine when and how to notify the user to brush their teeth more.

*Paper authored and research conducted as part of CS 6762 - Signal Processing, Machine Learning, and Control at the University of Virginia School of Engineering and Applied Sciences, taught by Felix Lin, PhD.

Although our preprocessing for model development had involved taking mean accelerometer values for each second, for our watch app, we chose to identify each recorded sample individually instead, for which there were variable samples per second. (The Asus ZenWatch 2 model uses variable frequencies for sampling based on power and screen status, but the threshold is between 50-200 Hz). This allowed us to experimentally modify the threshold of data examined to alter how the watch behaved without having to retrain a new model each time. Ultimately, we classified each second of data as brushing if 5% of the samples in that timeframe are classified as such. This threshold seemed to be best for immediately identifying brushing the second that brushing occurred, but it seemed to fail with our full-app testing.

For our full app, upon counting 30 seconds of tooth brushing, the app assumes the user is actually brushing their teeth, as opposed to making a similar but more brief gesture. From this, the app enters a state where it monitors for brushing to stop, which is indicated by 10 consecutive seconds of non-brushing. In this case, unless the user has already brushed for the recommended amount of time (2 minutes), the application issues a vibration and visual notification urging the user to continue brushing, as shown in Figure 1. From this, the app re-enters a state of assuming the user is no longer brushing their teeth.

We chose the above intervals because we wanted our app to be able to handle inconsistencies with toothbrush detection. Since it is unlikely for people to brush for less than 30 seconds, we chose 30 seconds as the starting condition. This should be an appropriate minimum length of time for discriminating it from similar activities of noticeably shorter lengths. Likewise, since it is unlikely for people to stop brushing for more than 10 seconds consecutively unless they are done, we chose 10 seconds as the stop condition. This also gives enough room for incorrect model predictions while ensuring the user is continuously brushing their teeth.

Fig. 1. Photo of running smartwatch sensing application on Asus ZenWatch 2.

III. RESULTS

Through sequential selection, X_acc_mean added 96.2264% to the accuracy, and Y_acc_mean added 0.7861% to the accuracy; no more features were added after these, since the next best feature, Y_acc_std, only added 0.1572%, which is less than 0.5%. From this, our decision tree had a classification accuracy of 97.0126% before being implemented onto the watch. All of these accuracies were calculated using 10-fold cross validation. Figure 2 shows the decision tree splits and Table I shows the confusion matrix from our testing.

This decision tree was successfully integrated into our application to sense toothbrushing. To verify our decision tree implementation, we started with a basic smartwatch app that vibrated when the application detected toothbrushing. This initial app was then updated into our final model, so the detect-then-immediate-buzz feature is no longer present.

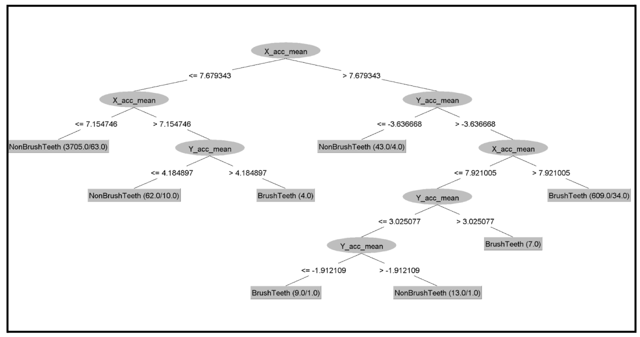

Fig. 2. Decision tree with X_acc_mean and Y_acc_mean as features.

Table I
CLASSIFICATION CONFUSION MATRIX

True Class \ Predicted Class	BrushTeeth	NonBrushTeeth
BrushTeeth	586	86
NonBrushTeeth	47	3733

We define qualitative testing as the evaluation of the full smartwatch application behavior, comparing it to desired results in a number of use cases. These use cases included: brushing teeth for a full 2 minutes, brushing teeth for less than 2 minutes, and a variety of non-teeth-brushing actions that could confuse our decision tree. The same subject that collected teeth-brushing data was used for testing, and the results are on the following page in Table 2, and they demonstrated that our current model is not as effective for sensing real-time tooth brushing as it is with pre-recorded (and trained-on) data.

Table 2
QUALITATIVE TESTING RESULTS

>2 minutes teeth brushing	Adequate success rate for detecting tooth brushing (>60 seconds) but high false positives and did not reach 2 minutes
<2 minutes teeth brushing	Adequate success rate for detecting tooth brushing (>30 seconds) but high false positives
Various other actions	No or very miniscule false positives (<2 seconds for each action)

IV. DISCUSSION

This project has a few functional drawbacks to its practicality, particularly with respect to how lifestyle and sensing methodologies conflict. Tooth brushing is a task commonly done early in the morning and later at night. Thus, many people may put on their smartwatches after brushing their teeth in the morning and take them off before brushing them in the evening; this completely prevents detection, which would then unnecessarily nudge them to brush more once they start wearing the watch, given the current design.

Additionally, our smartwatch was worn on the non-dominant hand for measurement, which is typical for most smartwatch users. To make this project work as a proof-of-concept, our user, both in data collection and testing, brushed their teeth with their non-dominant hand. However, toothbrushing is typically done with the dominant hand. This means that data could have been faulty or not characteristic of brushing, since the user might have been brushing more awkwardly. This also means that, with a more valid approach, putting the smartwatch on the opposite wrist of the user as the toothbrush, the sensor on the non-dominant hand would entirely miss key hand movements associated with the teeth-brushing action.

Finally, as with many machine learning projects, especially those that are integrated into a mobile application, robust testing of that integration is necessary. To verify our application, our project would have benefitted from more app-specific tests and iteration based on those results. In addition to this, just as with individual machine learning models, our application's success could improve through collecting more real-world data, such as on many more everyday activities. This would provide more non-brushing activities (and more realistic ones) for our model to better discriminate brushing from.

Overall, we believe that our methods and results demonstrate a framework for sensing workflows applicable to everyday tasks, which represents the breadth of future implementations developers can look towards. One especially interesting avenue for further work is sensing directly from toothbrushes, which could provide more tailored sensing data and effective responses. This also addresses the above issue of sensing on the opposite hand. Furthermore, the fact that the same smartwatch model was used for both measuring training data and app performance evaluation should be considered when looking at successes. Further research should evaluate the device interoperability of the decision tree model and consider how different sensing methods may impact success rates. Additionally, we consider user-specific model training a potential avenue for future research, as the exact toothbrush and corresponding motions likely differ across users.

V. CONCLUSION

Our application and corresponding decision tree had a success rate of 97.0126% for classifying any given second as brushing or non-brushing. Our full-app, qualitative testing showed that real-world implementation was far more difficult, as there was a large gap between expected and actual results regarding sensed tooth brushing rate. Additionally, we noticed that holding (as opposed to wearing) the watch led to a higher detection rate, indicating the potential that a smart toothbrush with embedded accelerometer sensors may have.

These results indicate an effective project that could encourage healthy behaviors but has significant room for real-use improvement through more data collection, fine tuning, and iterative development. Future work should be dedicated to diversifying signal types, further model tuning with consideration of different user characteristics, and an evaluation of user response to application feedback. A holistic, user-in-the-loop development and model training cycle would not only improve this work but help extend this research into smart health applications.