Hydration Monitoring System for Everyday Use

GEETHIKA HEMKUMAR, University of Texas at Austin, USA

1 INTRODUCTION

Maintaining proper hydration is vital to a person's energy levels, health, and overall well-being. Many bodily functions, including nutrient transport and body temperature regulation, are dependent on the body maintaining proper hydration [5]. Remembering to drink water consistently throughout the day in the midst of a busy day at school or work can be difficult. Hence, having an application that will monitor a person's hydration and prompt them to drink water throughout the day would be extremely beneficial.

The goal of my project is to develop a mobile application that works in concert with a wearable device to track a user's hydration and improve their hydration habits. The goal of enabling hydration tracking can be accomplished by integrating a model to recognize hydration events into the mobile application. The latter goal, improving hydration habits, can be accomplished by tracking/recognizing a variety of factors related to hydration/dehydration. Such factors include: physical activity, ambient temperature, and the amount of time since the last drinking event.

The results of the activity and drinking gesture recognition models, the focus of this work, are promising, with ten-fold cross-validation accuracies of 93.457% for the best activity recognition model and 95.390% for the best drinking gesture recognition model. Further work is needed to improve the generalizability of the models to unseen participants, but the initial results of this project show promise towards the development of a mobile application to improve hydration habits.

1.1 Contributions

The key contributions of the work are:

(1) A drinking gesture recognition model to track hydration events
(2) An activity recognition model to track physical activity
(3) The framework for a hydration monitoring and reminding iOS application that receives IMU and temperature data from a wearable device over Bluetooth Low Energy (BLE) in order to recognize drinking gestures and schedule hydration reminders

Though many of these ideas have been explored before in prior work, to my knowledge there is no work that combines all of them together. My project places an emphasis on developing a hydration monitoring application that can be used on a daily basis, which to my knowledge has not been explored extensively previously.

2 PRIOR WORK

The prior work that is closest to my proposed project is AutoHydrate [5]. This work developed a wearable system for monitoring hydration levels. The system consisted of a microphone placed near a person's throat for detecting drinking activity (a binary classifier is used to classify drinking vs non-drinking activity), a smartwatch for classifying physical activity, a computing device to process this data, and a smartphone app to aggregate the insights from this data into information about a person's hydration levels. An equation based on medical insights is used to give regular hydration recommendations based on food/fluid intake, activity level, and age/weight/height. Similar to AutoHydrate, I built a classifier to recognize various everyday physical activities. Additionally, I used similar activity classes that AutoHydrate used. Unlike AutoHydrate, I used a wrist-worn device that streams IMU data over BLE to an iOS application that hosts a model to recognize drinking gestures.

Author's Contact Information: Geethika Hemkumar, geethika.hemkumar@utexas.edu, University of Texas at Austin, Austin, Texas, USA.

There are a few existing works that utilize physiological signals to estimate hydration levels. A recently published paper utilized a smartphone camera to record a person's fingertip [1]. Photoplethysmography (PPG) signals are extracted from these videos and are used to create both a binary classifier to determine dehydration and a multi-class classifier to determine a person's hydration level on a scale from 1 to 4. A less recent paper utilized features extracted from electrocardiogram (ECG) signals as input to a SVM trained to classify volume depletion/dehydration [3]. Another work used features extracted from a person's galvanic skin response, a measure of skin conductance, in order to train several classifiers to predict hydration level [4].

There is a previous work that detects drinking gestures using inertial data collected from a smartwatch [2]. The paper divides drinking gestures into "sip" and "fetch" motions, where the former is a shorter motion and the latter encompasses the full drinking motion. Additionally, the paper utilizes features regarding a person's posture while drinking from a container in order to recognize the container type and the amount of fluid in the container. This paper is a very comprehensive study on drinking gesture detection.

Finally, there is a hydration monitoring system in the market called HidrateSpark[1]. This is a smart water bottle instrumented with a sensor to track liquid consumption that connects to a smartphone.

3 APPROACH

(a) Hardware

(b) Wireframe of iOS Application

Fig. 1. Hardware for the Project (Left) and Wireframe for the Hydration Monitoring Application (Right)

3.1 Hardware

Figure 1a shows the hardware used for the project. The top left part of the image is a prototype of the wrist-worn device used to collect drinking (and other wrist-based) gesture data. The prototype consists of an Arduino Nano 33 BLE Sense Rev2 with headers[2] powered by 2 CR2032 batteries placed in a coin cell holder attached to the bottom of the purple enclosure. The bottom right part of the image shows an iPhone X loaded with the custom iOS app. The app receives IMU and temperature data from the Arduino board using BLE. The software for the app is documented in the following section.

3.2 Software

Figure 1b shows a wireframe of the hydration monitoring iOS application. I did not have time to implement the *Consumption History* and *Hydration Status* views completely, but believe that this will be trivial compared to the scope of the work that has been completed.

The user-facing views are the *Consumption History*, *Hydration Status*, and *Connect Device* views. The *Connect Device* view would allow users to connect their Bluetooth device to the app to allow data streaming. The

[1]https://hidratespark.com/

[2]https://store-usa.arduino.cc/products/nano-33-ble-sense-rev2-with-headers

Consumption History view would provide a daily, weekly, and monthly view of a user's activity and hydration history, allowing them to understand how to improve their hydration habits. The *Hydration Status* view would provide real-time data, including: the number of drinking events in the last hour, the current temperature, and a "hydration status" indicating whether or not the user is drinking enough water.

3.3 Data Collection

Data for the activity recognition model was collected by each participant on their own phone using the Sensor Logger[3] app. Participants collected accelerometer and gyroscope data of four activities: walking, resting, climbing stairs, and running. All participants collected data for the former two activities, and some additionally collected data for the latter two (these were optional to ensure participants did not do activities that they felt physically unable to do at the time).

Two datasets were collected for the gesture recognition model. For both datasets, each participant performed five drinking events in a semi-scripted manner. The two datasets varied in terms of the activities between drinking events and the other gesture data that was collected. For all data collection episodes (across all labels), accelerometer and gyroscope data was collected.

The first dataset, termed dataset/strategy 1 for the remainder of the paper, had two labels: "drinking" and "resting." The drinking and resting data was collected as one recording. In the iOS app, I had a toggle button to label when the participant was performing a drinking gesture and when they weren't. During a "drinking" phase, participants would perform a water-drinking gesture, and during a "resting" phase, participants would perform non-drinking activities such as: walking around, pretending to type, and sitting at a table in a resting position.

The second dataset, termed dataset/strategy 2 for the remainder of the paper, had four labels: "drinking", "resting", "typing", and "piano." As with data collection for strategy 1, the drinking and resting data was collected as one recording. The main difference between strategy 1 and strategy 2 is in the activity in the data labelled as "resting." In strategy 2, participants rested their wrists on a table during the "resting" phase instead of performing other activities. In collecting the second dataset, I wanted to investigate whether or not training the model on additional data with labels unrelated to drinking water, instead of grouping all non-drinking activities into one label as with dataset 1, would improve the model's ability to discriminate between drinking and non-drinking events.

Importantly, the drinking gesture data collected is from drinking events with a wide variety of water containers, including: mugs, tumblers, and water bottles of various shapes and sizes.

3.4 Model Training

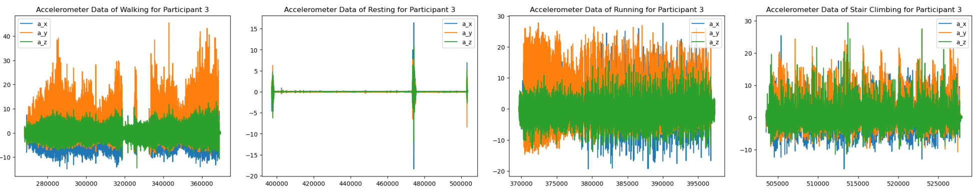
3.4.1 Data Preprocessing. For data preprocessing, I extracted frames for each accelerometer and gyroscope data recording. Each frame size was 100 samples and the hop length was 50 samples. All data in a frame was grouped under the same label.

Sensor	Features Extracted
Accelerometer	mean, variance, energy, and RMS of the signal for each axis
Gyroscope	mean, variance, energy, and RMS of the signal for each axis

Table 1. Features Extracted for Each Sensor

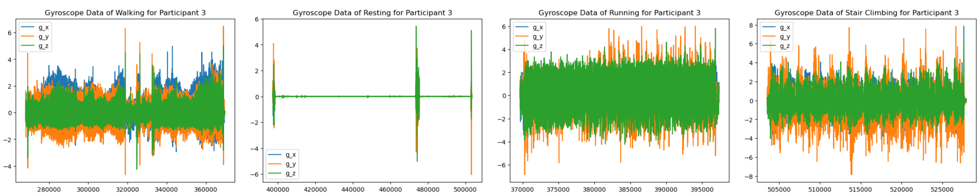
3.4.2 Features Extracted. Table 1 shows the features extracted from the accelerometer and gyroscope data.

[3]https://github.com/tszheichoi/awesome-sensor-logger



(a) Walking (b) Resting (c) Running (d) Stair Climbing

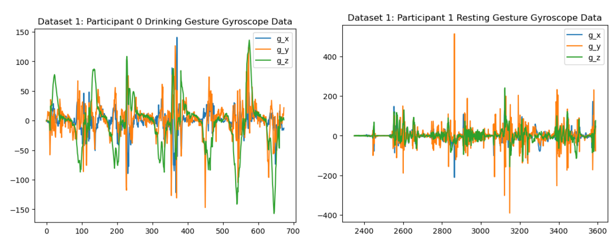
Fig. 2. Accelerometer Data (x, y, and z axes) from Participant 3 for All Four Activities



(a) Walking (b) Resting (c) Running (d) Stair Climbing

Fig. 3. Gyroscope Data (x, y, and z axes) from Participant 3 for All Four Activities

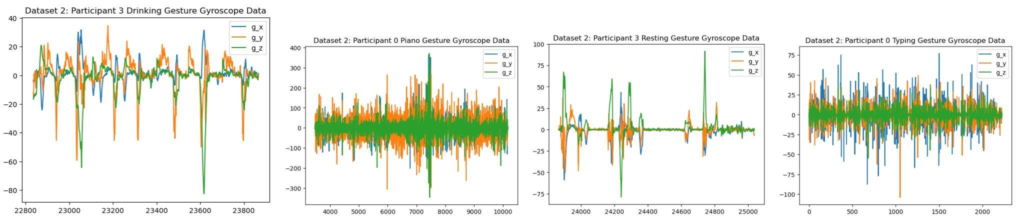
3.4.3 Activity Recognition Model. Section 4.1 describes the models used for activity recognition. Visually, there is a noticeable difference between the accelerometer and gyroscope data across activities, as shown in Figures 2 and 3, which provides strong motivation for using a classifier for activity recognition.



(a) Drinking Participant 0 (b) Resting Participant 1

Fig. 4. Gyroscope Data (x, y, and z axes) For Drinking and Resting Activities From Gesture Dataset 1

3.4.4 Drinking Gesture Recognition Model. Sections 4.2 and 4.3 describe the models used for drinking gesture recognition. Visually, there is a noticeable difference between the accelerometer and gyroscope data across activities. The gyroscope data is particularly interesting, as the plots of the gyroscope data for various wrist-based activities show a distinct cyclical pattern that varies across activities, as shown in Figures 4 and 5 (there is a visual difference between the accelerometer data across activities as well, but the visualization of the data is harder to understand intuitively). This visual intuition provides strong motivation to use a classifier for drinking gesture recognition. As a note, "drinking gesture recognition model" is synonymous with "gesture recognition model" throughout this paper.



(a) Drinking Participant 3 (b) Piano Participant 0 (c) Resting Participant 3 (d) Typing Participant 0

Fig. 5. Gyroscope Data (x, y, and z axes) from Participants 0 and 3 for All Four Activities From Gesture Dataset 2

3.5 Integration of Models with Mobile Application

I exported the activity recognition and both drinking gesture recognition models (one trained on gesture dataset 1 and the other trained on gesture dataset 2) as ONNX[4] files. ONNX is a library that allows cross-platform deployment of machine learning models. I didn't get a chance to integrate these models into my application, but I plan to do this in the future, further explained in Section 6.

4 RESULTS

The results are split into two main sections. First, results related to the activity recognition model are discussed, and then results related to the gesture recognition model. For both models, the following analyses are performed:

(1) Ten-fold cross validation across data from all participants
(2) Leave-one-participant-out (LOPO) cross validation across all participants

For (1), the models used are: a random forest classifier with 100 trees, a SVM with RBF kernel, and a MLP classifier with the maximum number of iterations set to 500. All classifiers used were from the *scikit-learn*[5] library. All displayed confusion matrices are the result of adding the confusion matrices across all folds together.

4.1 Activity Recognition Model

I collected activity data from 8 participants to evaluate the activity recognition model. The participants were required to perform the walking and resting activities for (at least) 5 minutes each, and were given the option to perform the stairs and running activities for 2-3 minutes. The following results are based on data from 7 of these 8 participants, as data from one of the participants was dropped due to sensor data synchronization issues.

To determine which model to use, I compared the performances (average precision/accuracy/recall across ten-fold cross-validation) of a random forest model with 100 trees, SVM kernel with RBF kernel, and MLP classifier with a maximum of 500 iterations. The random forest model yielded the best performance.

Next, I compared the peformances (precision/accuracy/recall based on the 80-20 train-test split approach) of three random forest classifiers trained on various combinations of the aforementioned accelerometer and gyroscope features. An "accelerometer features only" model was trained only using all extracted features from the x, y, and z axes of accelerometer data. A "gyroscope features only" model was trained only using all extracted features from the x, y, and z axes of gyroscope data. Finally, a "both accelerometer and gyroscope features" model was trained using all extracted features from all axes of both the accelerometer and gyroscope data. The random forest model trained on features from all axes of both the accelerometer and gyroscope data yielded the best performance.

[4]https://onnxruntime.ai/
[5]https://scikit-learn.org/stable/

Based on the above two evaluations, the random forest classifier with 100 trees trained on both the accelerometer and gyroscope features (across all axes for both) is used for the remainder of analysis with the activity recognition model.

Average Accuracy	Average Precision	Average Recall
93.457%	93.419%	93.457%

Table 2. Average Accuracy, Precision, and Recall of Ten-Fold Cross-Validation Using Activity Data From All Participants

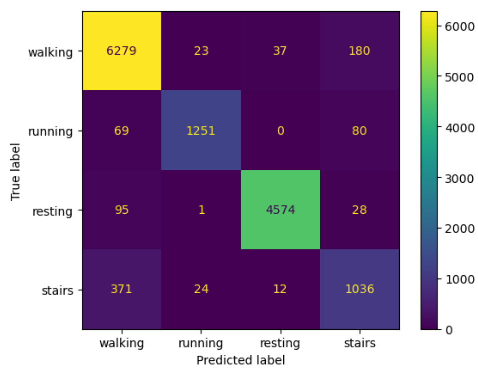


Fig. 6. Combined Confusion Matrix From 10-Fold Cross-Validation Using Activity Data From All Participants

4.1.1 Ten-Fold Cross-Validation Across All Participants. Table 2 shows the average accuracy, precision, and recall scores for ten-fold cross-validation on the aforementioned random forest model using data across all participants. These results are quite promising, demonstrating that the model can discriminate between the various activities reasonably well. Figure 6 shows the corresponding confusion matrix, the results of which corroborate the findings in Table 2.

4.1.2 Leave-One-Participant-Out (LOPO) Cross-Validation Across All Participants. Figure 7a plots the accuracy, precision, and recall scores from each fold of leave-one-participant-out (LOPO) cross-validation using data from all participants. The accuracy plot (red) overlaps almost completely with the blue plot (recall). Excluding participants 4 and 5, these scores are reasonable. Participant 5 had recently sprained their ankle, hence they were not able to perform the climbing stairs and running tasks to the full potential. Though the complete explanation for the significant difference in accuracy/precision/recall for participant 4 is unknown, it is likely that the differences are due to variations in data collection conditions (i.e., where the phone was placed when data was being collected).

Figure 7b shows the corresponding confusion matrix. The majority of the confusion is between the stairs and walking classes. I believe that this confusion can be mitigated if more data for the stair climbing activity was collected.

4.2 Drinking Gesture Recognition Model: Strategy 1

To evaluate strategy 1 of implementing the drinking gesture recognition model, I collected data from 7 participants. The participants were asked to perform five drinking events each, each in a way that felt natural to them. Between drinking events, they performed a variety of everyday tasks, such as getting up from their

(a) Per-Fold LOPO CV Scores (b) Combined LOPO CV Confusion Matrix

Fig. 7. Line Plot of Accuracy, Precision, and Recall Scores From Each Fold Of LOPO CV (Left) and Combined Confusion Matrix From LOPO CV (Right), Both Using Activtiy Data Across All Participants

chair, miming typing gestures, and resting at their table. The following results are based on data from all of these participants.

To determine which model type and features to train the model on, I performed a similar analysis to that in Section 4.1. Based on this, a random forest classifier with 100 trees trained on both the accelerometer and gyroscope features (across all axes for both) is used for the analysis of strategy 1 of the drinking gesture recognition model.

Average Accuracy	Average Precision	Average Recall
86.472%	86.599%	87.896%

Table 3. Average Accuracy, Precision, and Recall of Ten-Fold Cross-Validation Using Strategy 1 Gesture Data From All Participants

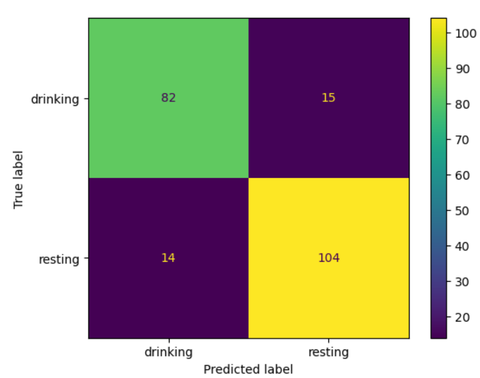
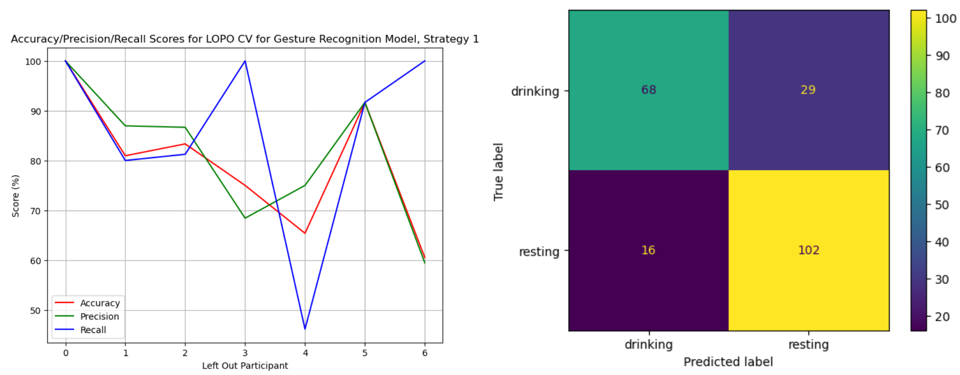


Fig. 8. Combined Confusion Matrix From 10-Fold Cross-Validation Using Strategy 1 Gesture Data From All Participants

4.2.1 Ten-Fold Cross-Validation Across All Participants. Table 3 shows the average accuracy, precision, and recall scores for ten-fold cross-validation on the aforementioned random forest model using data collected using strategy 1 across all participants. These results are promising, demonstrating that the model can discriminate between drinking and non-drinking activities somewhat well. Figure 8 shows the corresponding confusion matrix, the results of which corroborate the findings in Table 3.



(a) Per-Fold LOPO CV Scores

(b) Combined LOPO CV Confusion Matrix

Fig. 9. Line Plot of Accuracy, Precision, and Recall Scores From Each Fold Of LOPO CV (Left) and Combined Confusion Matrix From LOPO CV (Right), Both Using Strategy 1 Gesture Data Across All Participants

4.2.2 Leave-One-Participant-Out Cross-Validation Across All Participants. Figure 9a plots the accuracy, precision, and recall scores from each fold of leave-one-participant-out (LOPO) cross-validation using data collected using strategy 1 from all participants. The results indicate that the model does not generalize very well across participants, as the performance is variable across folds.

Figure 9b shows the corresponding confusion matrix. When looking at the combined results from LOPO cross-validation (shown by the confusion matrix), it appears that the model is overall able to distinguish between drinking and resting events somewhat well, but could the model could benefit from additional data to improve its generalization capabilities.

4.3 Drinking Gesture Recognition Model: Strategy 2

To evaluate strategy 2 of implementing the gesture recognition model, I collected data from 6 participants. The participants were asked to perform five drinking events, each in a way that felt natural to them. Between drinking events, they rested their wrists on the table. Additionally, each participant separately collected 2-5 minutes of either piano playing or typing data. The following results are based on data from all of these participants.

To determine which model type and features to train the model on, I performed a similar analysis to that in Section 4.1. Based on this, a random forest classifier with 100 trees trained on both the accelerometer and gyroscope features (across all axes for both) is used for the analysis of strategy 2 of the gesture recognition model.

4.3.1 Ten-Fold Cross-Validation Across All Participants. Table 4 shows the average accuracy, precision, and recall scores for ten-fold cross-validation on the aforementioned random forest model using data collected using strategy 2 across all participants. These results are quite promising, demonstrating that the model can discriminate between drinking and non-drinking activities quite well. Figure 10 shows the corresponding confusion matrix, the results of which corroborate the findings in Table 4.

Average Accuracy	Average Precision	Average Recall
95.390%	95.777%	95.390%

Table 4. Average Accuracy, Precision, and Recall of Ten-Fold Cross-Validation Using Strategy 2 Gesture Data From All Participants

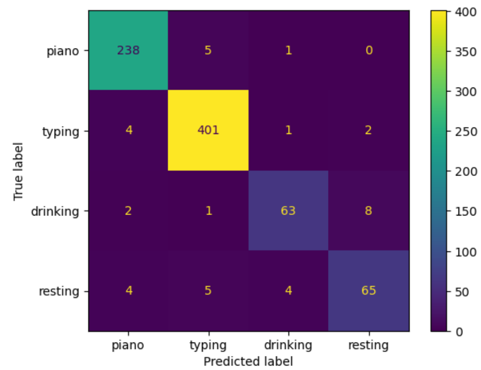
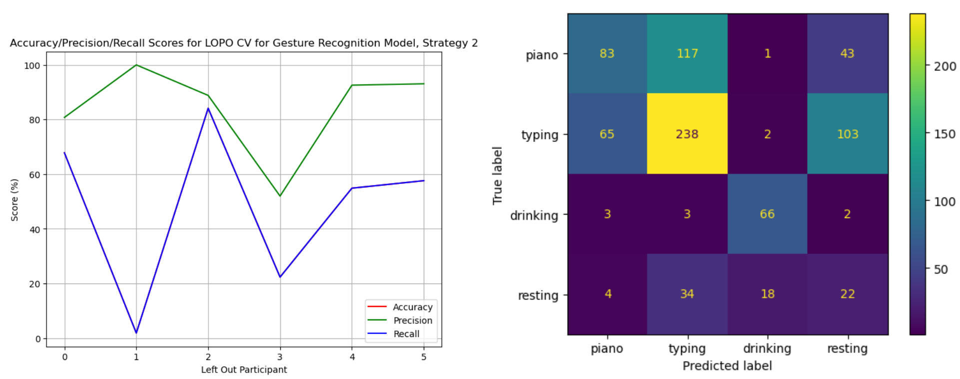


Fig. 10. Combined Confusion Matrix From 10-Fold Cross-Validation Using Strategy 2 Gesture Data From All Participants



(a) Per-Fold LOPO CV Scores (b) Combined LOPO CV Confusion Matrix

Fig. 11. Line Plot of Accuracy, Precision, and Recall Scores From Each Fold Of LOPO CV (Left) and Combined Confusion Matrix From LOPO CV (Right), Both Using Strategy 2 Gesture Data Across All Participants

4.3.2 Leave-One-Participant-Out Cross-Validation Across All Participants. Figure 11a plots the accuracy, precision, and recall scores from each fold of leave-one-participant-out (LOPO) cross-validation using data collected using strategy 2 from all participants. The accuracy plot (red) overlaps almost completely with the blue plot (recall). The results indicate that the model does not generalize very well across participants, as the performance is quite variable across folds.

Figure 11b shows the corresponding confusion matrix. The majority of the confusion is between the piano and typing classes. Intuitively, this makes sense, as gyroscope data of the piano and typing gestures look very similar as shown in Section 3.4.4 (the accelerometer data of the piano and typing gestures look quite similar too, but I did not include it for brevity). There is little confusion among samples in the drinking class, indicating that the model can recognize data from this class quite well.

5 DISCUSSION

This project demonstrated the possibility of creating a mobile application to monitor users' hydration and improve hydration habits using drinking gesture detection and activity recognition models. There are a few important limitations to address.

Firstly, Section 4 shows that the test accuracy of the activity and drinking gesture recognition models are reasonably high when trained on data from all participants. However, the results of leave-one-participant-out (LOPO) cross-validation suggest that the models are not as generalizable to data from unseen participants compared to data from those in the training set, as accuracies are quite variable across folds. Collecting more drinking gesture data in a variety of scenarios would improve the model's generalizability to unseen participants. Further, collecting "in-the-wild" data would also improve the model's accuracy.

Another limitation is that the classes in the drinking gesture recognition model trained with strategy 2 is limited. Adding more wrist-based gesture classes would improve the model's ability to perform robustly in-the-wild, where people perform a wider variety of wrist-based activities other than just typing, playing piano, and drinking water (some other examples are writing, cooking, and cleaning).

6 FUTURE WORK

An important area of future work is to enable the drinking gesture recognition model to estimate the volume of water consumed in a given drinking event to improve the quality of hydration monitoring.

Another area of future work is to integrate the activity and gesture recognition models with the iOS application, which is also discussed in Section 3.5. The results of the activity and drinking gesture recognition models are central to the core logic of the iOS application. For example, inferences from these models would be used to update the drinking event and physical activity history in the *Consumption History* page described in Section 3.2. Additionally, the number of drinking events in the last hour on the *Hydration Status* page, also described in Section 3.2, would be updated based on inferences from the drinking gesture recognition model.

The frequency at which the app would remind users to hydrate is based on a user's activity history, ambient temperature, and drinking event history. The logic for encoding the frequency of reminders based on the aforementioned factors is another area of future work.

Finally, further work should be done to investigate the impact of handedness on the drinking gesture recognition model's accuracy. The majority of the participants I collected data from were right-handed, and thus I did not have enough data to do a thorough analysis on the impact of wearing the device on the left hand on model performance. I believe that the model can generalize well to either a left or right handed configuration if a sufficient amount of data is collected with both left and right handed users.

7 CONCLUSIONS

This project presented a promising step towards creating a mobile application to track hydration events and improve hydration habits by reminding users to hydrate at a variable frequency based on various factors. These factors include: physical activity, the amount of time since the last hydration event, and ambient temperature. My project developed a drinking gesture recognition model to track hydration events and an activity recognition model to track physical activity. The results of evaluation of these models showed that while the model generalizes well to data within the participant pool it was trained on, some further, more diverse and in-the-wild data collection is needed to enable the model to generalize to unseen participants.

REFERENCES

[1] Rose Alaslani, Levina Perzhilla, Muhammad Mahboob Ur Rahman, Taous-Meriem Laleg-Kirati, and Tareq Y Al-Naffouri. 2024. You can monitor your hydration level using your smartphone camera. *arXiv preprint arXiv:2402.07467* (2024).

[2] Oliver Amft, David Bannach, Gerald Pirkl, Matthias Kreil, and Paul Lukowicz. 2010. Towards wearable sensing-based assessment of fluid intake. In *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. IEEE, 298–303.

[3] Anthony Kaveh and Wayne Chung. 2013. Classification of hydration status using electrocardiogram and machine learning. In *AIP Conference Proceedings*, Vol. 1559. American Institute of Physics, 240–249.

[4] Sidrah Liaqat, Kia Dashtipour, Kamran Arshad, and Naeem Ramzan. 2020. Non invasive skin hydration level detection using machine learning. *Electronics* 9, 7 (2020), 1086.

[5] Yehenew Mengistu, Minh Pham, Ha Manh Do, and Weihua Sheng. 2016. AutoHydrate: A wearable hydration monitoring system. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 1857–1862.