The Ohio State University, Department of Computer Science & Engineering

Smartphone-Based Intelligent System: Training AI to Track Sobriety Using Smartphone Motion Sensors

Jackson A Killian[1], Kevin Passino[2], Arnab Nandi[1], Danielle R Madden[3], John Clapp[3]

1- The Ohio State University, Department of Computer Science and Engineering; 2- The Ohio State University, Department of Electrical and Computer Engineering; 3- University of Southern California, Department of Social Work

INTRODUCTION

- **Alcohol consumption is significant cause of death** globally [1].
- Concentrated risk on college campuses (binge drinking).
- **Weekly mobile-based interventions are effective** in reducing consumption in students [2].
- **Holy Grail: real-time mobile intervention** to prevent events like:
  - Drunk driving
  - Alcohol Poisoning
  - Violence
- **No existing method for tracking sobriety in real-time** at scale…
- **So no studies** about real-time mobile interventions
- **We built an AI system to passively track a user's sobriety using only motion data from a smartphone.**

PRIOR WORK

- Mobile AI for tracking sobriety is an new/open field
- Recent work is promising but insufficient

Table 1: Studies training mobile AI to track sobriety

Study	Size	Reporting Method	Input Data	Accuracy/ Correlation
Arnold et al. [3]	**N = 6**	**Self Reports**	Accelerometer	70.0%
Bae et al. [4]	N = 38	**Self Reports**	**Private/ Personal**	96.6%
Gharani et al. [5]	N = 10	**Self Reports**	Accelerometer	R ≥ 0.9

- Each study takes 2 steps forward, 1 step back via:
  - Small sample size
  - Using next-day self reports to establish sobriety (bias)
  - Tracking/Storing myriad personal information

OUR DATA

- Gathered **N = 20** students during senior bar crawl
- Used **Transdermal Alcohol Content (TAC) Sensors**
  - TAC tracked through skin ~= BAC
  - Sampled every 30 minutes for 24 hours
- Stored **only accelerometer data**
  - Sampled at 40Hz for 12 hours
- Allowed for:
  - Unbiased and data-rich results
  - Downstream system that
    - Requires no personal data
    - Operates in real time

METHODS

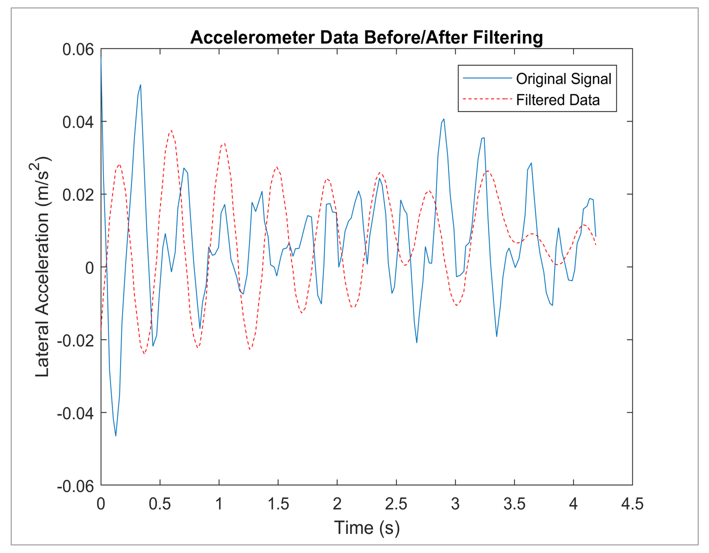

Figure 1: Accelerometer reading before and after filtering.

1. Gather Data
   - Built smartphone app to collect motion data
2. Clean and Segment
   - Raw data is **noisy**
   - Accelerometers are **low-cost**
   - TAC sensors **harried by environment**
     - Splashes of alcohol
     - Clothing between skin/sensor

- **Filter with MATLAB** signal processing Toolbox
- Apply **Low Pass Filter** – Chebyshev II
- Remove high frequency **data that lacks biological meaning**
- Subtract 45 min. from TAC to match BAC [ref]
- Segment into **windows of length λ**

3. Extract Features
   - Previous studies:
     - Mean/Variance of signal
     - **Spectral Features**
     - Size/Length of Stride
   - New Feature Set:
     - **Capture variance of gait** within a window
     - Chop to 4 smaller snippets
     - Bin frequency content per snippet
     - Accomplished with 13 "bins" of Mel Cepstral Frequency Coefficients (**MFCC**)
     - Have 13x4 matrix M ➔ Take covariance $M*M^T$

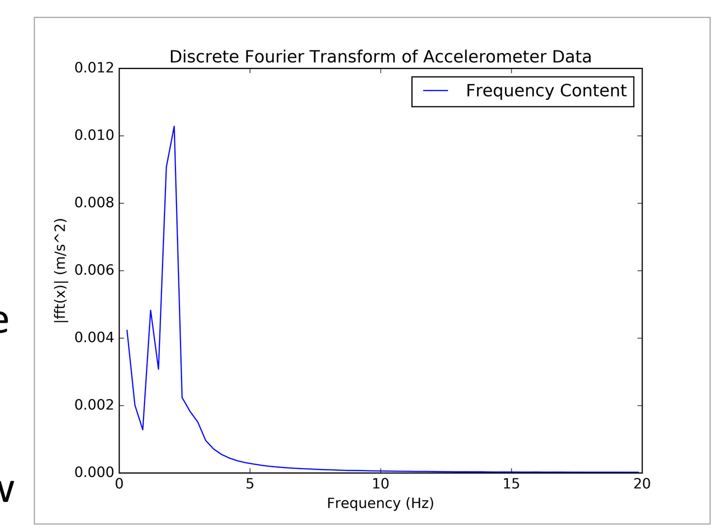

Figure 2: Frequency content of a window of accelerometer data. Used to calculate **spectral features.**

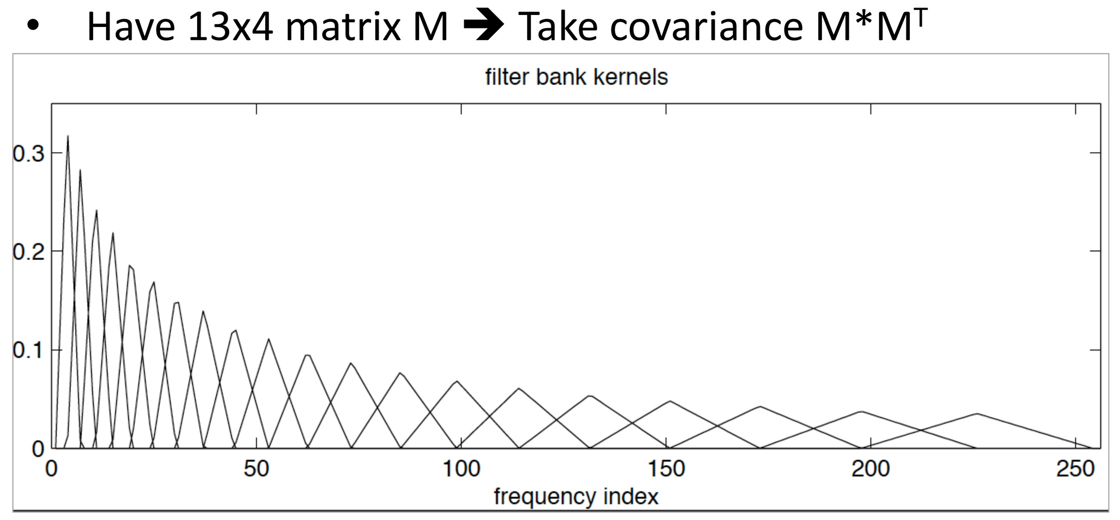

Figure 3: Visualization of how MFCCs are calculated. Each area under a triangular region represents a bin under which the given frequency content would be summarized.

4. Build and Train Classifiers

Table 2: Classifiers we used and why

Classifier	Description	Reasoning
Convolutional Neural Network	Deep neural network - alternates filtering and subsampling of raw data	Recent success classifying time-series data (speech recognition)
Long Short-Term Memory Network	Recurrent network that keeps a state or "memory" over a period of time	Successful in time-series prediction. Intuitive (bartender example)
Multilayer Perceptron Net	Basic neural network with one hidden layer	Versatile, commonly used in classification tasks
Random Forest	Takes mode of predictions of set of randomly generated decision trees	Versatile, quick to train, robust (especially against overfitting)
Support Vector Machine	Uses optimization/kernel trick to find exact decision boundary (even nonlinear)	Versatile, commonly used. Project into non-linear feature space.

RESULTS

We trained each machine to make binary classifications of "sober" or "intoxicated", parameterizing the cutoff as α. **Figure 4** shows the results of 3-fold cross validation for many values of α and λ.

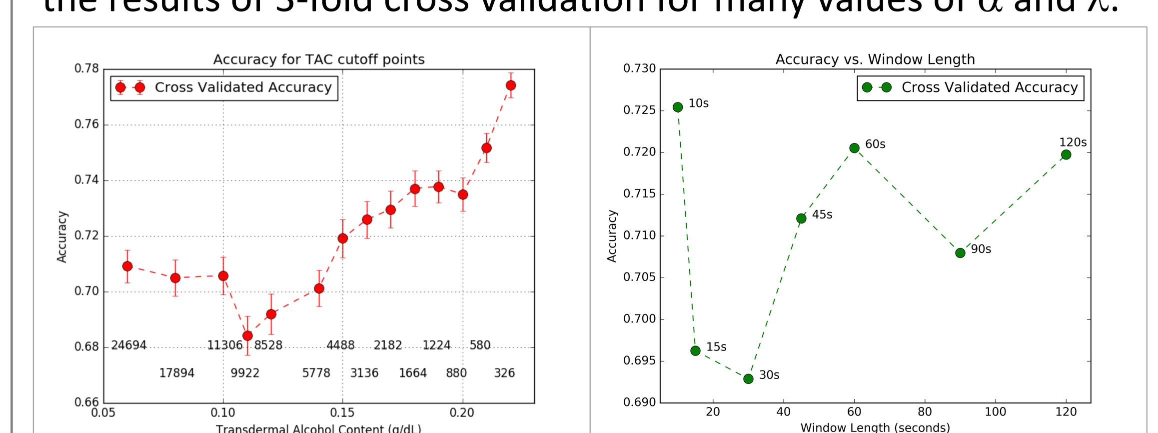

Figure 4: Accuracy vs. TAC cutoff α (left), accuracy vs. window length λ (right). Gains for $\alpha \geq 0.14$ came with tradeoff of reduced sample size: we used $\alpha = 0.08$ (legal limit) and $\lambda = 10$s.

Our final dataset had **26,087 rows** of data each with 243 features. For each machine, we ran 3-fold cross validation to establish its optimal parameters. Each row was labeled "sober" or "intoxicated" if the TAC was below or above α respectively.

Table 3: Per-class and overall accuracy by classifier

Classifier	Accuracy	Sober-Accuracy	Intoxicated-Accuracy
CNN	68.20%	-	-
LSTM	70.15%	-	-
MLP	72.5%	**81.62%**	53.27%
Random Forest	74.20%	77.43%	**68.45%**
SVM	**75.04%**	81.54%	61.49%

- Entries are average test accuracy from 3-fold splits
- **MLP** network best at identifying **sober data**
- **Random Forrest** best with **intoxicated data**
- **SVM** performed **best overall**

Table 4: Classifying power added by MFCC feature set

Classifier	Accuracy w/ MFCC	Accuracy w/o MFCC	Difference
MLP	73.49%	65.81%	7.68%
Random Forest	74.20%	67.63%	6.57%
SVM	75.04%	66.97%	**8.07%**

To calculate the classifying power added by our new MFCC feature set, we retrained and tested the 3 best machines on data without those features. **Table 4** shows that our new feature set adds up to an **8% gross increase** in accuracy.

CONCLUSIONS

- Gathered a **high-quality data set**
  - Large sample (**26,087 rows**, N = 20 participants)
  - Accelerometer-only (**lightweight, no privacy issues** downstream)
  - TAC sensors for ground truth (**no bias**)
- Achieved **best known binary classification accuracy**
  - SVM achieved **75.04%** accuracy
- Discovered a **powerful new feature set** for this domain
  - MFCC coefficients added up to **8% gross increase** in classification power

FUTURE WORK AND IMPOVEMENTS

- **Could not control for phone placement**. Should separate by pant pocket, purse, shirt pocket, etc
- **Did not isolate only walking data** – this could improve results
- **All classifiers better at identifying sober data** – suggests further inquiry needed to understand difference between sober/intoxicated
- **We will build mobile app to house the trained classifiers**
  - Healthy users **monitor drinking habits**
  - **Cheap, real-time tool** for researchers
  - Trained on rich, unbiased, real-world field data ➔ **reliable**
  - With relative performance, get **as high as 77% accuracy**.

BIBLIOGRAPHY

1. Rehm J, Mathers C, Popova S, et al. Global burden of disease and injury and economic cost attributable to alcohol use and alcohol-use disorders. Lancet. 2009;373(9682):2223-33.
2. Suffoletto B, Kristan J, Chung T, et al. (2015) An Interactive Text Message Intervention to Reduce Binge Drinking in Young Adults: A Randomized Controlled Trial with 9-Month Outcomes. PLoS ONE 10(11): e0142877. https://doi.org/10.1371/journal.pone.0142877
3. Arnold Z, Larose D, Agu E. Smartphone inference of alcohol consumption levels from gait. Healthcare Informatics (ICHI), 2015 International Conference on. IEEE. 2015. 417-426. DOI: 10.1109/ICHI.2015.59.
4. Bae S, Ferreira D, Suffoletto B, Puyana JC, Kurtz R, Chung T, Dey AK. Detecting Drinking Episodes in Young Adults Using Smartphone-based Sensors. Conference: PACM Interact. Mob. Wearable Ubiquitous Technol. (IMWUT), At Maiu, Hawaii. 2017;1(2):5. DOI: 10.1145/3090051.
5. P. Gharani, B. Suffoletto, T. Chung, et al. An artificial neural network for movement pattern analysis to estimate blood alcohol content level, Sensors 17 (12) (2017) 2897.

THE OHIO STATE UNIVERSITY

ACKNOWLEDGEMENTS

We thank the Arts and Sciences Honors Committee for their generous funding and support of this thesis work. We also thank Srinivasan Parthasarathy and Jiongqian Liang for their guidance regarding data preprocessing and machine learning methods.