Characterizing Drinks through WiFi analysis

Anette Habanen, Karl Raud
Institute of Computer Science, University of Tartu, Tartu, Estonia

Abstract

People are mostly good at differentiating between different liquids. Color, transparency and fizziness are few of the ways that can be used for it. But if these attributes are the same for two drinks, then it is not so easy for a human to know. That is why it is important to test out if pervasive data science methods (more specifically WiFi signals) could be use to detect differences between liquids. In our project we trained a model to do live predictions for various drinks.

Introduction

The easiest method is to just look at the drinks. Coca-cola and Fanta are very easily distinguishable just based on a color. But if you have a milk in front of you, it is difficult to say if it is a lactose-free one or not. For these purposes, there has to be other ways to differentiate between liquids. If humans can not determine the difference by themselves, other tools can be used. For example ultrasonic waves (1) and x-rays (2) can be used.

These methods either do not work in every situation or they are expensive and hard to replicate. Because of that we wanted to try a pervasive method to solve this problem. One solution would be to use WiFi signals to characterize the drinks. A person could be wearing a ring that transmits WiFi signal that goes through the drink and is received by their phone. Phone could then analyze the received signal and give out a prediction about the type of drink the person is holding.

Dataset

Collecting Data

To collect data, we needed a way to create WiFi signal and to collect it. To do it we had two little devices. One of them, called ‘access point’, sent out the WiFi signal. The other one, called ‘station’, received the signal and saved it. Station was connected to a computer that was using a program ESP-IDF, that saved the WiFi signal time series information into a .csv file. The saved result is **Channel state information** (CSI) data, along with other metadata, such as signal strength, timestamps and other channel information.

For our experiment, a cup with liquid inside it was placed between the access point and station. The devices were placed very close to (almost touching) the cup to have them as close to the liquid as possible. With this setup, 1 minute of collecting data gave us about 650 datapoints. We collected data for four liquids: water, plum juice, yogurt and beer. We chose these drinks because they are quite different from each other. For every drink we collected about 16 minutes of training data and 5 minutes of validation data.

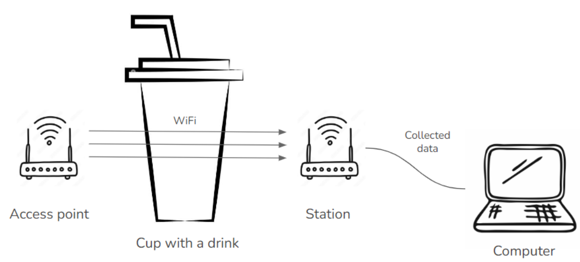


Figure 1. Figure of the data collecting setup.

Preprocessing Data

After gathering the data, it needed some preprocessing. First, we removed some unnecessary data. The signal info in the .csv file consisted of an array with 128 features of CSI data. Some of the features of the CSI data do not carry important information and can be removed to reduce noise. We selectively removed 32 redundant features, so we were left with 96 CSI features.

We also formatted the data in a way that we look at our data in sequences. One datapoint consists of 100 consecutive measurements. Different datapoints can have overlapping sequences of measurements. That way, we have more data for training the model.

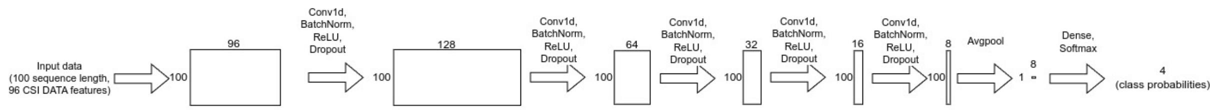


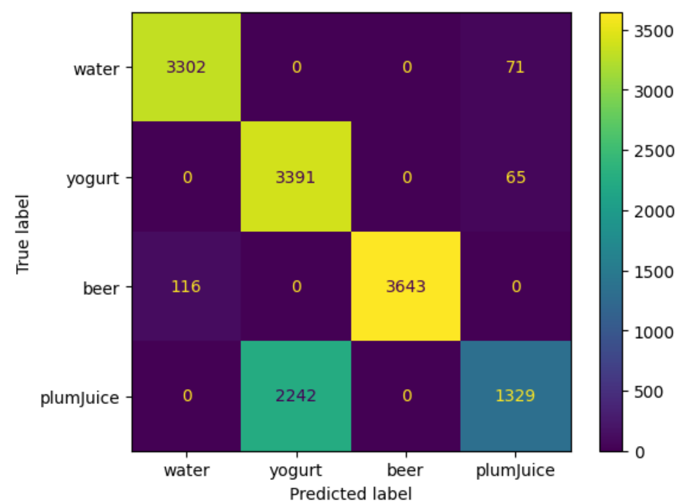
Figure 2. Model architecture for learning signal type per drink.

Model

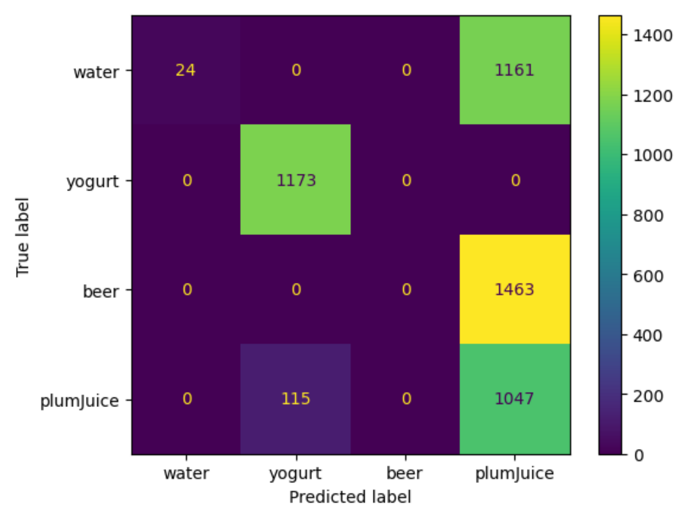
We wanted to train a model that would take in the preprocessed data and would learn the specific signal type that represents a drink. To do that we trained a custom convolutional model for time series classification. This model consisted of 1D convolution, batch normalization, ReLU and Dropout layers as seen in Figure 2. The custom convolutional model was trained with a batch size of 4096, l2 regularization of 0.1, dropout 0.5, with Adam optimizer, learning rate 0.001, for 300 epochs. The model was validated on a different dataset, captured about 30 minutes later of the train dataset.

We also tried classification with other models, like Rocket transforms (3), Catch22 transforms (4), HIVE-COTE 2.0 (5) and Self-attention based models (6). But it was found that these models were either too costly to run, or did not have good enough results mostly due to overfitting.

Here is the Github repository with the final convolutional model architecture.



(a) Train and validation data captured within the exact same conditions.



(b) Train and validation data captured within similar conditions.

Figure 3. Confusion matrices for drink prediction.

Results

Model Results

As seen from Figure 3(a), the model can learn to predict drinks with about 80% accuracy, if the train and validation data are taken in the exact same conditions. The only varying factor is the drink.

Figure 3(b) depicts the confusion matrix, if train and validation data are gathered on different days. This means, that the environment has changed slightly more. For example, position of items in the room, surrounding other WiFi signals, even moisture (7) can affect the recieved CSI data. Because of these additional varying factors, the model got 45% accuracy within similar conditions, in contrast to the 82.4% accuracy, gotten with exact same conditions.

Live Predictions

Our end goal was to train a good model that would make these predictions in real time. For this we took our model, connected our testbed to a computer and let the model do the predictions live. For consistency, we used an ensemble of 10 seperately trained models, instead of one, to account for variabilities during training. If we put a drink between the devices, we could inspect the models predictions. For the model to explain its predictions, it also outputs confidence scores for each drink. Softmax can be applied to these confidence scores, in order to get percentages.

Here is the video, demonstrating drink predictions in real time.

Lessons Learned

Even though we were able to train a model that could do live predictions, collecting data for this type of task is not easy. We collected a lot of different datasets and here are the lessons that we learned from each of them.

Dataset 1 - indoors data with foil box

For our first dataset we took a cardboard box, covered it with foil and placed the cup, access point and station in the box. We did this so that other WiFi signals wouldn't interfere our signals. After training a model we found out that these signals are very sensitive. When we collected data for water class, we didn't seal the box the same way as with the other class. That way it was really easy to predict water.

Dataset 2 - outdoors data

To eliminate the chance that other WiFi signals would affect the data, we also went outside to a spot where there were no WiFi signals. There we collected a lot of different data regards to the locations between the cup and the devices. Afterwards when we looked at the data we realized that the inside and outside data are very different from each other. Training a model with both datasets in it gave back a lot worse results than training with these datasets separately.

Dataset 3 - inside dataset with other WiFi signals

At this point we were starting to think more about live predictions and how we should collect a dataset for it. This meant that we had to collect a dataset where we did not put the cup and the devices into a box and had them freely on the table. This way we could ensure that data about every drink is collected in the most similar environment. We collected this type of data on two separate days. Even though we used the same location and setup for both of these, this data again was different from each other. This showed us even more how sensitive this data is and how minuscule changes actually matter when collecting data.

Conclusions

We were able to train a CNN model that can classify drinks using WiFi signals. Based on this trained model we predicted results in real-time. We found that these WiFi signals are very sensitive and creating exactly the same conditions is almost impossible. To actually use this type of differentiation model, it would need to be generalised by collecting substantially more data in very diverse environments.

References

[1] McClements, D. J., and Gunasekaran, S. (1997). Ultrasonic characterization of foods and drinks: Principles, methods, and applications. Critical Reviews in Food Science and Nutrition, 37(1), 1–46. https://doi.org/10.1080/10408399709527766

[2] Orghêda L. A. D. Zucchi, Silvana Moreira, Marcos J. Salvador, and Leandro L. Santos Journal of Agricultural and Food Chemistry 2005 53 (20), 7863-7869 DOI: 10.1021/jf0510945

[3] Dempster, Angus, François Petitjean, and Geoffrey I. Webb. "ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels." Data Mining and Knowledge Discovery 34.5 (2020): 1454-1495.

[4] Lubba, Carl H., et al. "catch22: CAnonical Time-series CHaracteristics: Selected through highly comparative time-series analysis." Data Mining and Knowledge Discovery 33.6 (2019): 1821-1852.

[5] Middlehurst, Matthew, et al. "HIVE-COTE 2.0: a new meta ensemble for time series classification." Machine Learning 110.11 (2021): 3211-3243.

[6] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017)

[7] Yang, Weidong, et al. "Wi-Wheat+: Contact-free wheat moisture sensing with commodity WiFi based on entropy." Digital Communications and Networks 9.3 (2023): 698-709.