Identifying Speaking and Drinking Events Within Audio Recordings for Multiactivity Analysis

Rethinking Ubiquitous Smart Sensing of Social Behaviour in the Wild

Dorothy Zhang

Supervisors: Koen Langendoen, Hayley Hung, Vivian Dsouza, Stephanie Tan

EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2024

Name of the student: Dorothy Zhang
Final project course: CSE3000 Research Project
Thesis committee: Koen Langendoen, Hayley Hung, Vivian Dsouza, Stephanie Tan, Qun Song

Abstract

Multiactivity analysis investigates one's coordination of actions within a social context, such as gestures and speech, usually using video recordings of the social activity, to further understand the rules of human behaviour. This paper focuses specifically on the coordination between speaking and drinking activities within a social setting, and explores the possibility of automatically identifying these events using audio captured from a drinking glass. As social interactions occur in vastly different contexts, this paper also investigates the effect that background noise might have on the accuracy of identifying these events. Different parameters and audio features were compared. Linear classification models LR and SVM with a linear kernel were able to achieve 100% accuracy for all sample lengths between 2 and 8 seconds using the first 20 PCA components from 60 audio features. The best performing feature in identifying speaking and drinking events was MFCCs, achieving an F_1 score of 99.4% on average across models with a training sample length of 3 seconds. Background noise had different effects on classification accuracy depending on the type, with music lowering the F_1 score to 74.3%, noisy room audio to 64.7%, and podcast audio simulating the presence of other speakers to 59.6% using MFCCs and a 3-second sample length.

1 Introduction

Multiactivity is a term used to describe one's coordination of multiple activities occurring at the same time. Compared to its close sibling — multitasking, where one might be eating dinner while watching a TV series, multiactivity focuses more on the coordination of multiple activities within a social setting [1], such as talking while driving with a passenger in a car or on the phone [2]. When more participants are involved, one's actions are affected by another's, and vice versa, stringing together orderly and meaningful interactions. Analyses on multiactivity focus on the sequential development of actions — why *this* happens at *that* time, to study the underlying rules and frameworks governing human communication [1].

As humans are not able to breathe in or out and drink at the same time [3], the act of speaking and drinking must be sequentially organized in some way. Current research on multiactivity make use of video recordings of social interactions to qualitatively analyse the coordination of verbal and physical expressions between participants [1, 2, 4, 5]. This paper looks into the feasibility of using only audio data to identify both speaking and drinking activities within social settings where it might be hard to set up overhead cameras to pick up such actions, such as large outdoor gatherings or low-light environments.

The main contributions of this paper are: investigating (i) whether it is possible to identify drinking and speaking events using audio captured from a drinking glass, (ii) how well the identification performs in different noisy environments.

2 Related Works

Hoey [4] closely analysed the coordination between speaking and drinking within social interactions, and gives insight into how the context of the interaction affects one's behaviour in drinking, or how other gestures are used to communicate instead of speech when one is occupied with drinking. Hoey's analysis is qualitative and micro-detailed, while this paper aims to take a quantitative approach and analyse coordination patterns on a more general scale with less focus on the conversation context. The automated identification of drinking and speaking events would allow for more efficient and large-scale analysis of multiactivity patterns between these activities.

Numerous studies have explored and achieved accuracies as high as 99% with detecting drinking events using inertial sensors attached to a drinking cup or one's arm [6–11], on the other hand, speaking events are harder to detect with the same type of data. Several studies using body-worn accelerators were only able to detect speaking status with an accuracy between 68% and 72% in standing social interactions [12–14]. Cabrera-Quiros et al. [14] specifically examined the relationship between gestures and speaking status, finding that speaking actions were not necessarily accompanied by gestures, hence the lower accuracy in predicting speaking status using gestural data. Attaching an inertial sensor on a drinking glass would not only suffer from the same problem, but would also be a difficult feat in social situations where drinking glasses are placed statically on the table in front of participants while speaking, as observed in Hoey's analyses [4]. In this case, the successful detection of any speaking action would rely on the sensor picking up sound vibrations from the cup. Considering these limitations, this paper chose to explore the feasibility of solely using audio data to distinguish drinking and speaking events.

Audio classification using machine learning is a well-researched topic. Many audio classification algorithms calculate audio features such as Mel-Frequency Cepstral Coefficients (MFCCs) for windows of a few milliseconds over the entire length of the audio sample, and then aggregated these values by taking the mean and variance to form a smaller feature vector for the entire audio piece [15]. As the main objective in this paper is to classify digital audio of speaking and drinking events, similar methods for generating feature vectors will be utilised.

Much existing research that detect or analyse swallowing sounds make use of a throat microphone [16–19], and extract frequency related features such as MFCCs or Fast Fourier Transforms (FFTs). These studies have been able to identify swallowing sounds from other noises originating from the throat at accuracy scores ranging from 93.7% [17] to 95.20% [18]. These results further motivate the focus on frequency features to identify swallowing sounds in this paper. The use of a throat microphone to capture swallowing sounds in a casual social setting could be considered rather intrusive and out of place, therefore, alternatively, this paper proposes the microphone to be placed on the drinking glass, simulating a possible smart drinking cup.

3 Experimental Setup

Due to the specific setup of the microphone being attached to the cup, no suitable existing audio datasets could be used, and custom data was collected for later experiments. These pieces of audio data are then processed according to the flowchart seen in Figure 1. After recording and annotating relevant audio clips, possible external noise could be added artificially to simulate the different environments in which a social interaction could take place. The feature vector is then constructed by calculating different audio features over multiple windows, taking statistical aggregations such as the mean, median, and variance, and then used in different classification models.

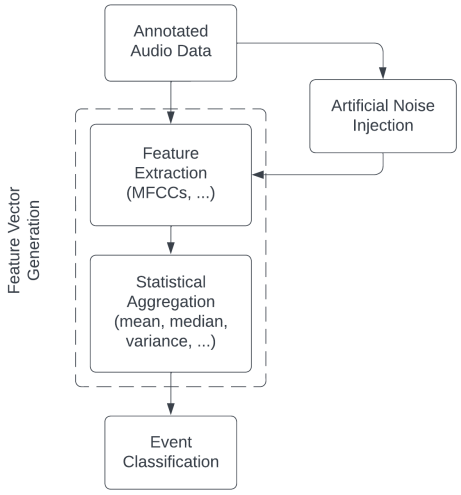


Figure 1: The processing for event classification using recorded audio data.

3.1 Data Collection

An Arduino Nano 33 BLE Sense was used to record audio data. This microcontroller board is equipped with a microphone capable of recording audio at 16kHz or 42kHz[1] at 16 bit digital resolution. A frequency of 16kHz was favoured to minimize the amount of data that needs to be processed while still having recordings of adequate quality, and the gain was set at the highest value.

To be able to pick up swallowing sounds as well as drinking sounds, the sensor was attached to a drinking glass using elastic bands to simulate a smart drinking cup. As previously mentioned, this setup was chosen to minimize intrusion from the sensing device. The sensor board is then connected to a laptop through a micro USB cable for power supply and data transfer. To transfer audio data through the serial port, a ring buffer code[2] was implemented on the board to allow for continuous sampling of audio data. The following types of audio were collected through the above described setup:

- Speaking: 12 audio clips of 9-10 seconds with the empty glass placed around 30cm away from the speaker, totalling to around 118 seconds in length.
- Drinking: 14 audio clips of 8-10 seconds, each clip having around 7 to 9 sips, totalling to around 126 seconds and 108 individual sips.
- Ambient noise: 12 audio clips of 10 seconds, totalling to around 120 seconds in length.
- Real-life social setting: 12 audio clips with 3 speakers in total and music playing in the background at times, totalling to around 47.5 seconds of speaking, 51 seconds of drinking, and 50 seconds of ambience.

To simulate the environments in which a social setting could occur in, different types of noise were also collected: classical music (Winter - Vivaldi[3]), a noisy room[4], and podcast audio[5] to simulate other speakers. These noises were played through the laptop speaker and recorded through the microphone to preserve any artefacts from the microphone and minimize artefacts that could come from downsampling the original audio to 16kHz, and then synthetically added using the Audiomentations library[6] with a minimum dB of -25.0 and maximum dB of -15.0. The music and noisy room audio were added to all audio classes to simulate different environments, and the podcast audio was only added to drinking and ambience audio to simulate other speakers. Figure 2 shows the spectrograms of one audio clip from each type of collected noise.

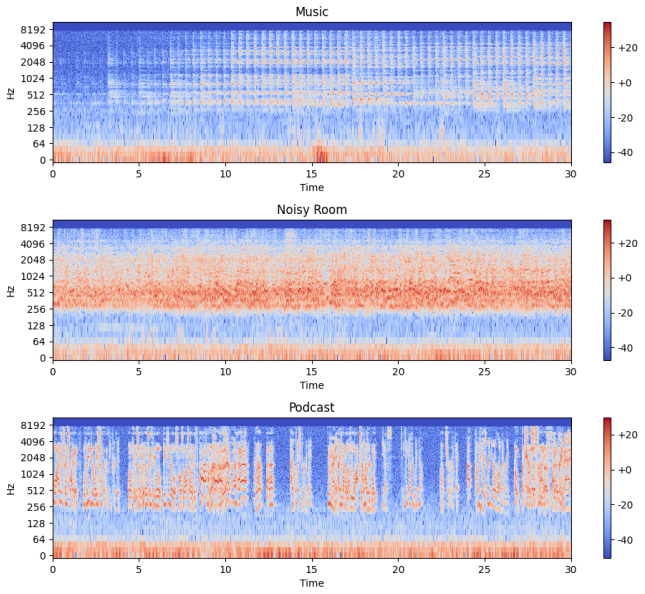
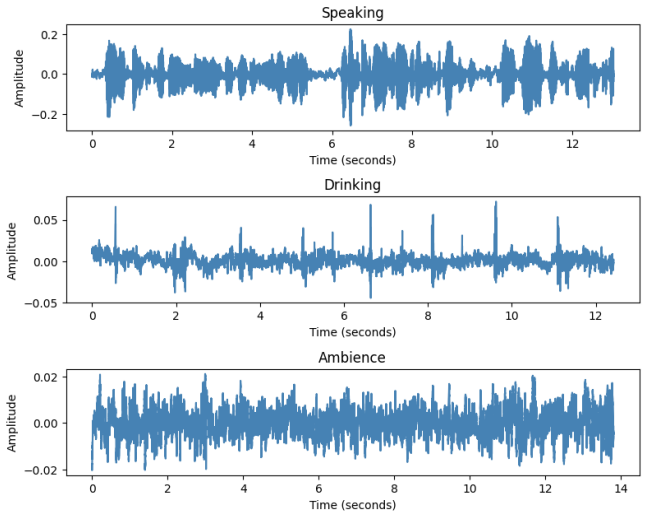


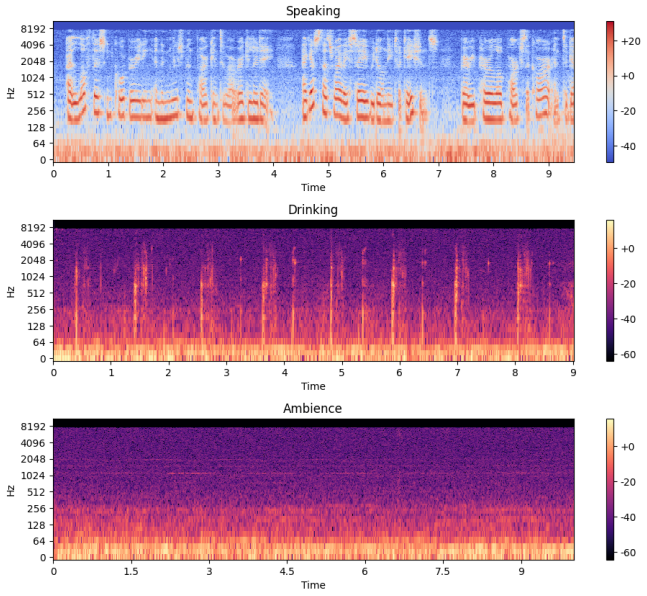
Figure 2: The spectrograms of collected noise.

[1]https://forum.arduino.cc/t/pdm-lib-issue-changing-the-sample-rate-pmd-begin/619569/2

[2]https://stackoverflow.com/a/76397224/14195852



(a)



(b)

Figure 3: (3a) An example of audio waveform of collected samples. (3b) An example of spectrograms of collected samples.

3.2 Feature Extraction

Figure 3a shows example audio waveforms of recorded audio clips from each class, and Figure 3b shows the frequency spectrum of the same audio clips. As there are noticeable differences across both amplitude and frequency characteristics for all three classes, both amplitude and frequency related features were extracted and compared. A total of 6 frequency related features and 1 amplitude related feature were chosen for comparison. These features were extracted using the audio processing library Librosa[7] with a frame size of 1024 and hop size of 256, and the mean, variance, median, minimum and maximum values were taken to form the feature vector, amounting to 60 features in total for training. The features were also separated into 3 distinct feature groups for performance comparison, and Figure 4 shows the first 2 components of their respective principal component analyses (PCAs) using a training sample length of 3 seconds. The following feature groups were extracted and compared:

- **Mel-Frequency Cepstral Coefficients** - As mentioned in earlier sections of this paper, MFCCs and FFTs were widely used in audio classification in general, as well as studies that analysed swallowing sounds with the use of a throat microphone. MFCCs represent the short-term power spectrum of a sound, and are calculated from FFTs. The FFT values calculated from small windows of the sound signal are mapped onto the mel scale that more closely approximates the human auditory system's response. More calculations are carried out before finally arriving at a set of coefficients. As there is a limited amount of collected data, only the first 6 MFCCs were taken to control the size of the feature vector accordingly. The resulting feature vector has 30 values.
- **Spectral Centroid**, **Spectral Contrast**, **Spectral Bandwidth**, **Spectral Roll-off** - These features are frequently used in music genre classification [20, 21], and provide insight into the timbre and textural properties of a piece of audio.
  - Spectral Centroid - Spectral Centroid represents the "centre of mass" of the spectrum.
  - Spectral Contrast - Spectral contrast measures the difference in amplitude between peaks and valleys in a sound spectrum.
  - Spectral Bandwidth - Spectral bandwidth measures the width of the spectrum, providing an indication of the range of frequencies present in a sound.
  - Spectral Roll-off - Spectral roll-off is the frequency below which a specified percentage (in this paper, 85% is used) of the total spectral energy is contained.

  These features form a feature vector of size 20.

[3] https://www.youtube.com/watch?v=TZCfydWF48c
[4] https://www.youtube.com/watch?v=UnhpCJ5tkW4
[5] https://wayneradiotv.podbean.com/
[6] https://github.com/iver56/audiomentations
[7] https://librosa.org/

- **Zero Crossing Rate**, **Root Mean Squared Energy** - These features give a basic overview of the changes in frequency and energy over the entire audio signal. They are simpler to compute and easier to understand in comparison with other features, and could be useful where computational resources are limited.
  - Zero Crossing Rate - ZCR measures the rate at which the signal changes sign. It captures information about the frequency content of the signal.
  - Root Mean Squared Energy - RMSE measures the overall energy or loudness of the signal. It provides information about the amplitude dynamics of the audio signal.

  These two features form a feature vector of size 10.

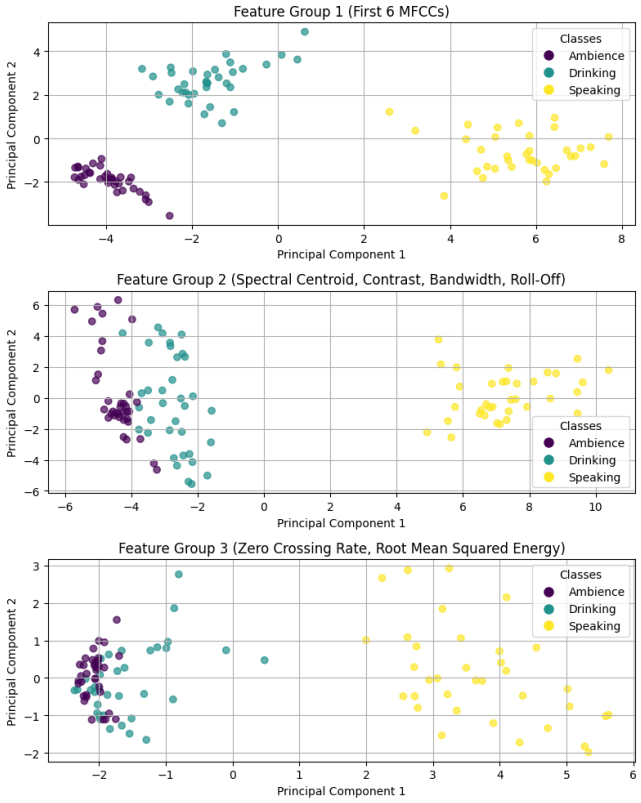


Figure 4: The first two components obtained from a PCA on all features, using a training sample length of 3 seconds.

3.3 Machine Learning Classification Models

To evaluate how well the features extracted can be used to distinguish the three different classes of speaking, drinking, and ambient noise, the library scikit-learn[8] was used to train 2 linear (LR and SVM) and 3 non-linear (KNN, DT, and RF) models, each having different strengths and weaknesses:

- **K-Nearest Neighbours:** KNN is a simple and straightforward learning algorithm that classifies a data point based on the majority class among its closest k neighbouring data points. In this paper, $k = 5$ neighbours are used. KNN is easy to understand and implement, and works effectively with small datasets. However, its speed decreases rapidly with larger datasets, as it has to execute distance calculations to all other points for every new data point.
- **(Multinomial) Logistic Regression:** LR is a statistical model that predicts the probability of a data point belonging to all classes and assigns it the most probable one. LR takes input features and combines them with coefficients that show how much certain features contribute to the probability of being in a certain class. These results are then put into a "softmax" activation function to derive the possibilities of all classes when more than 2 classes are present. The biggest advantage of LR is the practicality and interpretability of its calculation results. The coefficients show the relationship between feature and classes, and the final class probabilities show probable ambiguity between classes, which could be more informative than classifiers that only output the final classification. The main limitation of LR is its assumption of linearity between the dependent variable (classification outcome) and independent variables (features), which might not accurately reflect the true underlying relationship.
- **Support Vector Machine (Linear Kernel):** SVM is a supervised learning model that constructs one or several hyperplanes that best separates classes in the feature space. By utilizing different "kernel tricks" that transform the input data to a higher-dimensional space, SVMs allow non-linear distributions to be separated linearly again. In this paper, a linear kernel is used. SVMs are versatile as the kernel function can be individually defined and tailored to different data distributions, and work well with larger feature vectors. The disadvantages of SVMs are that they can be computationally expensive for large datasets, and perform less well with overlapping classes.
- **Decision Tree:** DTs are a non-parametric supervised learning method used for classification and regression. An optimized version of the CART algorithm is used in the scikit-learn library. The algorithm tries to continuously draw "splitting points" on the most significant features that distinguishes all the classes. DTs are "white box" models as its decisions are easy to understand and interpret like LR, however are also prone to overfitting — splitting the data points unnecessarily specific and narrow.
- **Random Forest:** RF is an ensemble learning method that constructs multiple DTs using different samples drawn with replacement from the training set. The predictions from these trees are then combined to reach a final decision. This is done to overcome some of the problems arising from DTs such as bias and overfitting. For this research, a number of 100 trees are used in the RF classifier. While RFs reduce the problems from DTs, they also diminish some of the benefits of using DTs.

[8] https://scikit-learn.org/stable/

Due to the use of a considerable number of trees in the learning process, RFs are less interpretable and computations are also more expensive.

3.4 Performance Evaluation Criteria

To compare and evaluate the accuracy of the different machine learning models, a stratified 5-fold cross validation with shuffling is used, along with the traditional scoring of recall (1), precision (2), F-measure (3), and weighted average score across classes (4):

$$\text{Recall}_e = \frac{\text{TP}_e}{\text{TP}_e + \text{FN}_e} \tag{1}$$

$$\text{Precision}_e = \frac{\text{TP}_e}{\text{TP}_e + \text{FP}_e} \tag{2}$$

$$\begin{aligned} \text{F}_{1e} &= \frac{2 \cdot \text{Precision}_e \cdot \text{Recall}_e}{\text{Precision}_e + \text{Recall}_e} \\ &= \frac{2 \cdot \text{TP}_e}{2 \cdot \text{TP}_e + \text{FP}_e + \text{FN}_e} \end{aligned} \tag{3}$$

$$\text{WeightedAvg}_{\text{score}} = \sum_{e \in E} w_e * \text{score}_e \tag{4}$$

Where for every type of event e, TP, FP, and FN refer to the number of correctly identified events, other events incorrectly identified as e, and event e being incorrectly identified as other events, respectively. The weighted average of a certain $\text{score} \in \{\text{Recall}, \text{Precision}, \text{F}_1\}$ is calculated by the sum of the products of w — the weight of each event in relation to the total dataset size, and the corresponding score of that event, for all events.

4 Results

The following results were observed for classifying speaking and drinking events with audio data using different parameters and altered training data.

4.1 Length of Training Data

Different lengths of samples were taken to train the machine learning models and their performances were compared. The samples are taken consecutively for each audio clip, and the remainder is discarded. Table 1 shows the number of samples that were extracted from all the audio clips for each sample length. As each audio clip is only between 8 and 10 seconds long, starting from a 6-second window and onward, the number of clips stays the same.

Class	Sample Length (seconds)							
	1	2	3	4	5	6	7	8
Speaking	116	57	35	24	21	12	12	12
Drinking	121	58	35	28	16	14	14	14
Ambience	120	60	36	24	24	12	12	12

Table 1: Number of samples per class when cut into samples of different lengths.

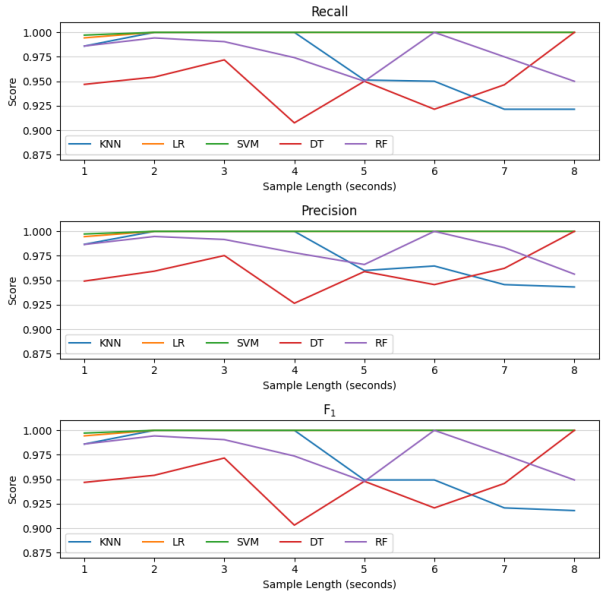


Figure 5: Weighted average recall, precision, and F_1 scores with different sample lengths, trained using the first 20 components derived from a PCA.

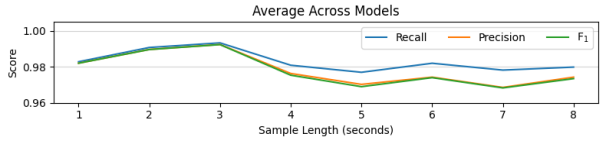


Figure 6: Average recall, precision, and F_1 scores across all models.

Figure 5 shows the weighted average scores across the three audio classes using the first 20 features, calculated from a principal component analysis (PCA) taking into account all 60 features across the three feature groups, trained on models KNN, LR, SVM, DT and RF. The score trend for all models except RF appears to increase starting from the 1-second window up until the 3-second window, and then decrease for KNN and fluctuate for DT. The scores for both SVM and LR reach 100% at the 2-second window and stay that way for the rest of the windows. DT fluctuates the most, reaching its highest F_1 score at the 8-second window and the lowest at the 4-second window, 100% and 90.3% respectively. KNN drops gradually after the 4-second window, reaching 91.8% at the 8-second window. These fluctuations in accuracies were only found in KNN, DT, and RF, and not in linear classifiers LR and SVM, suggesting overfitting in non-linear models. Figure 6 shows the average precision, recall and F_1 scores across all models, and a 3-second window performs the best, with an average F_1 score of 99.2%.

4.2 Feature Performance Comparison

Figure 7 shows the F_1 scores of the 3 feature groups in comparison with the first 20 features obtained from PCA with a training sample length of 3 seconds. MFCCs perform the best across all the models, only lowering to 97.0% on DT at worst.

Zero Crossing Rate (ZCR) and Root Mean Squared Energy (RMSE) perform the worst on average, ranging from 79.1% on DT to 88.5% on SVM. The spectral features reach as high as 100% on LR and SVM, and as low as 92.3% on DT. Table 2 shows the average score across all models. MFCCs contain 30 features, outperforming PCA, which has 20 features, only by 0.2%.

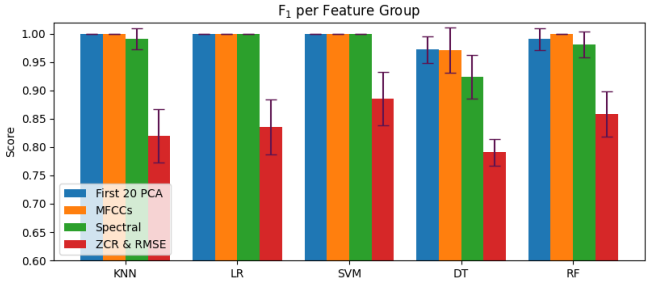


Figure 7: F_1 score for all feature groups using a sample length of 3 seconds.

	PCA	MFCCs	Spectral	ZCR & RMSE
Mean F_1	0.992	0.994	0.979	0.838

Table 2: Average F_1 score per feature across models.

4.3 Noise Injection

Figure 8 shows the F_1 score after different types of noises were synthetically added to the audio and classified using MFCCs with a training sample length of 1 second. Table 3 shows the average F_1 score across all models using different training sample lengths. The best performing scores are now at the 1-second and 2-second windows, instead of the 3-second window found before. Looking back at the classification score of MFCCs in Table 2, these scores show that additional noise can have quite an effect on the classification accuracy, decreasing the score by as much as 39.8% from 99.4% to 59.6% with the same features and sample length when recordings of a podcast are used to simulate the presence of other speakers. It is also interesting to note that even though both music and noisy room audio were added to all audio clips, music has a lesser negative effect on the classification accuracy.

Figure 9 shows the confusion matrices of the SVM classifier with a training sample length of 1 second with different types of noise. It is clearly demonstrated that speech of the drinking person can still be reliably classified across all types of noise, even when the voice of other speakers are added, while the main confusion now lies in distinguishing drinking audio from ambience. This shows that the addition of noise can disrupt the effectiveness of MFCCs features for the drinking class, possibly due to its frequency features being weaker in comparison to the added noise compared to that of speech, as seen in Figure 2 and Figure 3b. This could also explain why there is more confusion to be seen in the noisy room compared to music, even though both were added to all audio clips with the same dB range.

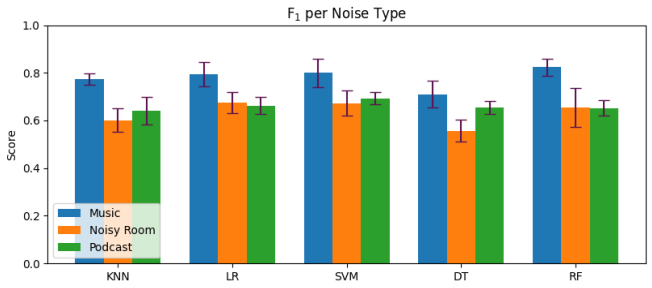


Figure 8: F_1 score for different types of noise with a training sample length of 1 second.

Length (sec)	Music	Noisy Room	Podcast
1	**0.772**	0.627	**0.661**
2	0.753	**0.650**	0.646
3	0.743	0.647	0.596
4	0.661	0.640	0.574
5	0.763	0.625	0.574
6	0.648	0.589	0.586
7	0.657	0.648	0.605
8	0.687	0.560	0.586

Table 3: Average F_1 score per noise type across models using different training sample lengths.

4.4 Training on Real-Life Data

Figure 10 shows the confusion matrices when real-life recordings are used as training data and tested on clean data using MFCCs at different sample lengths of 1 second, 2 seconds, and 5 seconds. The real-life data contains several noise sources such as music and other speakers. Overall, all models appear to be able to distinguish between the audio classes quite clearly, apart from several confusions especially between drinking and ambience which can be seen in KNN, DT, and RF models. LR and SVM are the most stable across different sample lengths, and all models seem to perform on the same level regardless of the increase in training sample length.

5 Responsible Research

Due to the nature of multiactivity analysis, it is unpreventable to have sensitive information from participants such as video or audio recordings. It is therefore always necessary to ensure proper communication and consent from participants of the potential use and implications of their data in research. For the data collection procedure, clean data was collected from just the author of this paper. For real-life audio concerning extra participants, approval was given from course staff through email to allow students within the same research group to participate in the recording. All participants involved in the data collection procedure were informed of the usage of their voice in further experimentation, namely that their voice would be recorded and later uploaded to a private

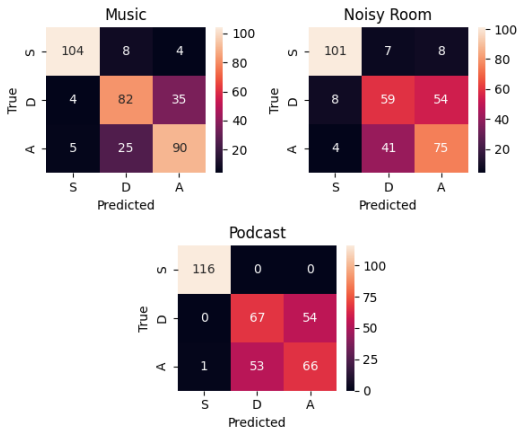


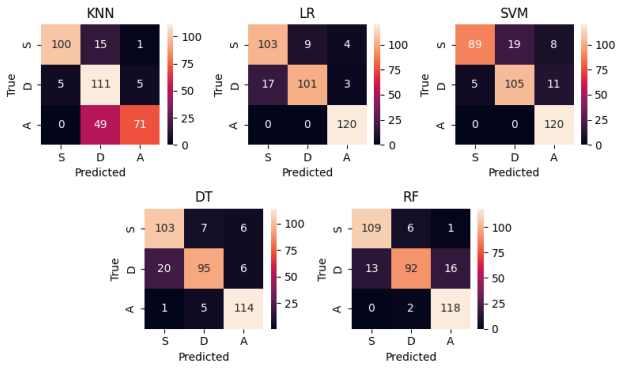
Figure 9: Confusion matrices for different types of added noise trained on SVM with a sample length of 1 second. (S) Speaking (D) Drinking (A) Ambience.

repository only the course administrators, the supervisors of our research team, and themselves would be able to access, and used in the process of training machine learning models to distinguish between speaking and drinking activities. The recordings were saved locally in .txt files labelled only as the corresponding audio type they were of, “speech”, “drink”, or “background”, along with other possible notes like “music” when music was present in the clip. These .txt files were then converted into .wav files for audio processing. Both the .txt and .wav files were only ever shared by uploading them to the research team’s private repository, and all local copies of the audio files will be deleted after the research project is over.

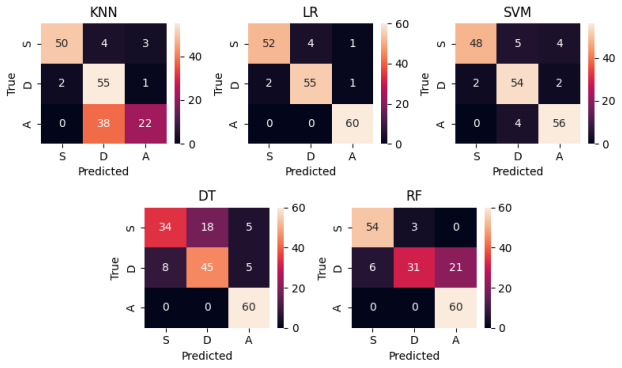
To ensure reproducibility of the experimental setup of this research project, all methods were explained in detail and parameters of features and machine learning models were also given. The original sources of the noises used, and various code libraries, were also linked in the footnotes. The code used to record audio from the Arduino board, to train machine learning models, and other code to process the data and generate figures were uploaded to the private repository. Within the repository, a README file has also been provided with guidelines on how to run the code written in individual Jupyter notebooks. Within the Jupyter notebooks, comments were added to explain the functionality of certain pieces of code and how to customize certain parameters.

6 Discussion

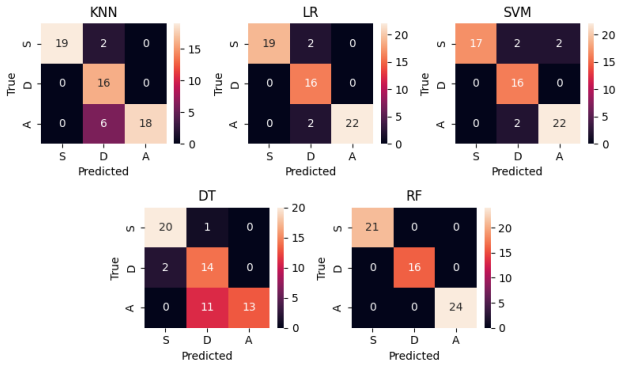
The experiments show that there are certain fluctuations in classification accuracy with different lengths of training samples. However, these could be inaccuracies resulting from the choice of parameters, or the decrease in size of the training data when longer samples were taken. For linear classifiers, a 2-second window is already enough to distinguish between clean audio of speaking, drinking, and ambience with 100% accuracy, while KNN and DT have an F_1 score fluctuating between 90% and 100%, and RF between 94% and 100% for all sample lengths. From these results it can be deduced that when dealing with clean audio, linear classifiers such as LR



(a)



(b)



(c)

Figure 10: Confusion matrices for classifying controlled data using training sample lengths of (10a) 1 second, (10b) 2 seconds, and (10c) 5 seconds. (S) Speaking (D) Drinking (A) Ambience.

and SVM with a linear kernel using sample lengths between 2 and 8 seconds is enough to fully distinguish the audio signals of speaking, drinking, and ambience.

The feature group performance comparisons show that while ZCR and RMSE performed the worst, averaging an F_1 score of 84%, could still be viable where computation resources are limited, as they are less resource intensive to calculate. Both MFCCs and the various spectral features also have great performance, scoring 99.4% and 97.9% on average respectively. As MFCCs are more complex to calculate than the spectral features, the minor difference in accuracy could make the latter a lot more favourable, especially in larger datasets.

The artificial addition of noise significantly affects the classification accuracy between audio classes, primarily making the audio of drinking harder to distinguishable from ambient noise, suggesting that the method of using audio to distinguish these events would be less feasible in noisy social environments. While multiactivity analysis often emphasizes the importance of conversation content [2,4] and typically avoids using noisy audio data, the method of identifying speaking and drinking activities can still be valuable in noisier environments. Speech from the drinking person is still identifiable when the presence of other speakers was simulated through adding audio recordings of a podcast, suggesting that MFCCs can distinguish between different voices and identify when the person of interest is speaking, when trained on their voice previously.

Audio from a natural social interaction was recorded with the initial intention to test how well models trained on clean data would perform with noisy data. The confusion matrices resulting from that experiment can be seen in Appendix 8 Figure 12. There was confusion between all classes, indicating that models trained on clean data were not generalizable to noisier audio. Looking at Figure 4, the audio classes are already very far separated from each other with the first 2 PCA components, so it's possible that the decision boundaries were underfitted for noisier audio. However, when trained on noisy audio, classification of clean audio was relatively accurate, demonstrating its generalizability in the other direction, indicating that training on noisy audio allows for more refined classification boundaries between the audio classes.

Using audio data alone for multiactivity analysis has limitations on its own. Without video data, it won't be possible to pick up details such as gestures, exchanges of gaze, and facial expressions, which are vital components that go into social interactions. These extra contexts will unfortunately not be present when only dealing with audio data, and the captured information is limited for further in-depth analysis.

This study has not investigated the performance of this activity recognition method with longer recordings of social activity, focusing instead on classifying individual audio clips. Additionally, the audio samples for speaking and drinking were collected from the same individual. Therefore, the method's viability still needs to be rigorously tested with diverse audio sources and dynamic real-life social settings to determine the generalizability of the results.

7 Future Work

This study has provided some insight into the classification of drinking and speaking events using audio data from clean and simulated noisy environments. Several suggestions for future research directions are outlined below:

As it became harder to distinguish drinking events in noisier environments, data from inertial sensors could be used in addition with audio data to detect drinking events more accurately. The microcontroller board used in this paper is equipped with such sensors, and as it is already placed on the drinking glass for audio recording, the same setup could be used to pick up motions of the glass as well. Several studies have investigated and achieved promising results using inertial sensors in a smart cup to detect drinking activity and other actions [6, 8, 10].

This study was not able to explore the performance of continuous activity recognition over longer audio recordings. Future research can further verify the feasibility of this method of activity recognition using metrics such as the one proposed by Ward et al. [22], examining its stability and reliability over longer periods of time.

8 Conclusion

This paper investigated the feasibility of using audio data to identify drinking and speaking events for multiactivity analysis. Three types of audio data were collected: speaking, drinking, and ambient noise. This was done through placing a microphone on a drinking glass to capture both swallowing sounds and speech of the drinking person. Various experiments show that when dealing with clean audio, using the first 6 MFCCs along with a sample length between 2 and 8 seconds yielded the best classification performance with linear classifiers, having 100% accuracy. However, when noises are introduced artificially to simulate different social environments, classification accuracy decreased by different amounts between 20% to 45% depending on the type of noise and training sample length. Future research can aim to improve accuracy by using inertial sensors in addition to audio data to distinguish drinking events from background noise, which was the primary source of confusion. The results obtained in this study have yet to be reproduced in more realistic social settings, but preliminary findings are promising and justify further investigation.

Appendix A

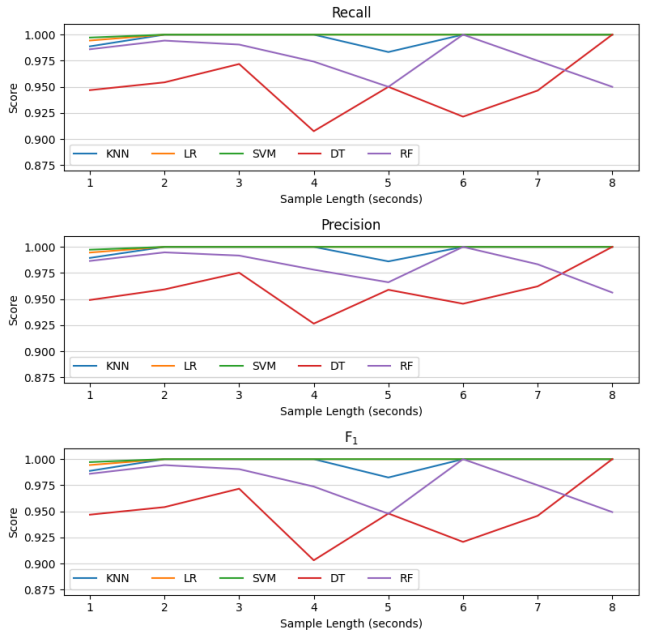


Figure 11: Weighted average precision, recall, and F_1 scores with different sample lengths, trained using the first 20 components derived from a PCA.

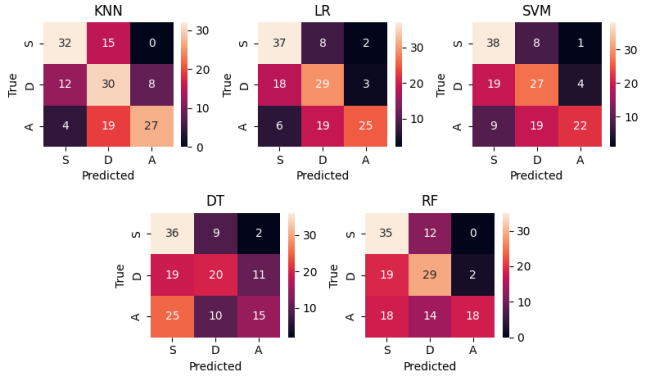


Figure 12: Confusion matrices for classifying noisy data using clean data with training sample length of 1 second. (S) Speaking (D) Drinking (A) Ambience.

References

[1] P. Haddington, T. Keisanen, L. Mondada, and M. Nevile, *Multiactivity in Social Interaction: Beyond multitasking*. John Benjamins Publishing Company, 2014. [Online]. Available: https://books.google.nl/books?id=cw1aBAAAQBAJ

[2] L. Mondada, "Talking and driving: Multiactivity in the car," *Semiotica*, vol. 2012, pp. 223–256, 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:55856386

[3] B. Martin-Harris, "Coordination of respiration and swallowing," *GI Motility online*, May 2006, publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/gimo/contents/pt1/full/gimo10.html

[4] E. M. Hoey, "Drinking for speaking: The multimodal organization of drinking in conversation," *Social Interaction. Video-Based Studies of Human Sociality*, vol. 1, no. 1, May 2018. [Online]. Available: https://tidsskrift.dk/socialinteraction/article/view/105498

[5] L. Mondada, "The methodical organization of talking and eating: Assessments in dinner conversations," *Food Quality and Preference*, vol. 20, no. 8, pp. 558–571, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950329309000329

[6] B. Dong, R. Gallant, and S. Biswas, "A self-monitoring water bottle for tracking liquid intake," in *2014 IEEE Healthcare Innovation Conference (HIC)*, Oct. 2014, pp. 311–314. [Online]. Available: https://ieeexplore.ieee.org/document/7038937

[7] T. Hamatani, M. Elhamshary, A. Uchiyama, and T. Higashino, "Fluidmeter: Gauging the human daily fluid intake using smartwatches," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 3, 9 2018. [Online]. Available: https://doi.org/10.1145/3264923

[8] M. Bobin, H. Amroun, M. Boukalle, M. Anastassova, and M. Ammi, "Smart cup to monitor stroke patients activities during everyday life," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2018, pp. 189–195.

[9] D. Gomes and I. Sousa, "Real-Time Drink Trigger Detection in Free-living Conditions Using Inertial Sensors," *Sensors (Basel, Switzerland)*, vol. 19, no. 9, p. 2145, May 2019. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6539019/

[10] K.-C. Liu, C.-Y. Hsieh, H.-Y. Huang, L.-T. Chiu, S. J.-P. Hsu, and C.-T. Chan, "Drinking event detection and episode identification using 3d-printed smart cup," *IEEE Sensors Journal*, vol. 20, no. 22, pp. 13 743–13 751, 2020.

[11] H.-Y. Huang, C.-Y. Hsieh, K.-C. Liu, S. J.-P. Hsu, and C.-T. Chan, "Fluid Intake Monitoring System

Using a Wearable Inertial Sensor for Fluid Intake Management," *Sensors*, vol. 20, no. 22, p. 6682, Jan. 2020, number: 22 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1424-8220/20/22/6682

[12] H. Hung, G. Englebienne, and J. Kools, "Classifying social actions with a single accelerometer," in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, ser. UbiComp '13. New York, NY, USA: Association for Computing Machinery, Sep. 2013, pp. 207–210. [Online]. Available: https://dl.acm.org/doi/10.1145/2493432.2493513

[13] E. Gedik and H. Hung, "Personalised models for speech detection from body movements using transductive parameter transfer," *Personal and Ubiquitous Computing*, vol. 21, no. 4, pp. 723–737, Aug. 2017. [Online]. Available: https://doi.org/10.1007/s00779-017-1006-4

[14] L. Cabrera-Quiros, D. M. J. Tax, and H. Hung, "Gestures In-The-Wild: Detecting Conversational Hand Gestures in Crowded Scenes Using a Multimodal Fusion of Bags of Video Trajectories and Body Worn Acceleration," *IEEE Transactions on Multimedia*, vol. 22, no. 1, pp. 138–147, Jan. 2020, conference Name: IEEE Transactions on Multimedia. [Online]. Available: https://ieeexplore.ieee.org/document/8734888

[15] J. Breebaart and M. F. McKinney, "Features for Audio Classification," in *Algorithms in Ambient Intelligence*, W. F. J. Verhaegh, E. Aarts, and J. Korst, Eds. Dordrecht: Springer Netherlands, 2004, pp. 113–129. [Online]. Available: https://doi.org/10.1007/978-94-017-0703-9_6

[16] H. Kalantarian, N. Alshurafa, M. Pourhomayoun, S. Sarin, T. Le, and M. Sarrafzadeh, "Spectrogram-based audio classification of nutrition intake," in *2014 IEEE Healthcare Innovation Conference (HIC)*, Oct. 2014, pp. 161–164. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7038899

[17] L. F. Santoso, F. Baqai, M. Gwozdz, J. Lange, M. G. Rosenberger, J. Sulzer, and D. Paydarfar, "Applying Machine Learning Algorithms for Automatic Detection of Swallowing from Sound," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Jul. 2019, pp. 2584–2588, iSSN: 1558-4615. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8857937?casa_token=QCE15b1qEw8AAAAA:nb08029S5wGHH00l4Czfd8Yx8Eu-IdwKEO8RiyiObZrkkDX2jMpz924g6GEVYsuVfLyouWQE

[18] S. Kimura, T. Emoto, Y. Suzuki, M. Shinkai, A. Shibagaki, and F. Shichijo, "Novel Approach Combining Shallow Learning and Ensemble Learning for the Automated Detection of Swallowing Sounds in a Clinical Database," *Sensors*, vol. 24, no. 10, p. 3057, Jan. 2024, number: 10 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1424-8220/24/10/3057

[19] H. Nakafuji, M. Imura, Y. Uranishi, S. Yoshimoto, and O. Oshiro, "Estimation of amount of swallowed water by analysis of swallowing sounds," *Japanese Society for Medical and Biological Engineering*, vol. 52, no. Supplement, pp. O–11, 2014, num Pages: O-12.

[20] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, Jul. 2002, conference Name: IEEE Transactions on Speech and Audio Processing. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1021072

[21] T. Pohle, E. Pampalk, and G. Widmer, "Evaluation of frequently used audio features for classification of music into perceptual categories," in *Proceedings of the Fourth International Workshop on Content-Based Multimedia Indexing*, 2005.

[22] J. A. Ward, P. Lukowicz, and H. W. Gellersen, "Performance metrics for activity recognition," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 1, jan 2011. [Online]. Available: https://doi.org/10.1145/1889681.1889687