ACTIVITIES OF DAILY LIVING DETECTION USING SMART MICROPHONES

ATHULYA SHAJI

MSC COMPUTER SCIENCE

SCHOOL OF COMPUTING

ULSTER UNIVERSITY BELFAST

B00873604

Contents

Abbreviations

1) AI – Artificial Intelligence
2) MLC – Machine Learning Classifiers
3) SVM – Support Vector Machine
4) KNN - k-Nearest Neighbors (kNN)
5) CNN – Convolutional Neural Network
6) LSTM – Long Short-Term Memory
7) ANN – Artificial Neural Networks
8) MFCC – Mel-Frequency Cepstral Coefficient
9) DNN – Deep Neural Network
10) HAR- Human Activity Recognition
11) ADLs– Activities of Daily Living

LITERATURE INVESTIGATION

Apart from the literature reviews included which were solely on audio classifiers, some of the other works which study about human activity recognition using different sensors are discussed below. Also works on parameter optimization for SVM and a detailed comparison of MLCs and NN in HAR are being considered during literature investigation.

1. Abnormal human activity recognition using SVM based approach

A. Palaniappan, R. Bhargavi and V. Vaidehi are presenting an Abnormal human activity recognition system in [1]. The system uses SVM. The authors detail the design of a multi-class SVM using a series of SVM classifiers together where each one predicts an output, the most positive value with be considered for labeling. The drawback of this approach is the increase in computational time. This issue has been addressed by introducing a transition table which helps decide which classifier to be used. The SVM classifiers need kernels for non-linear learning. The quadratic kernel is the kernel used in this project. Finally, the performance of the developed classifier is evaluated using metrics such as sensitivity, accuracy, and precision. The proposed SVM model gave an accuracy of 94.4%, with 96.7% precision, and 94.8 % sensitivity.

2. Feature Selection and Hyperparameter Optimization of SVM for Human Activity Recognition

The project [2] is building an SVM model for the recognition of human activities. The dataset used in this project is from the University of Southern California (USC). The activities considered are walking, sitting, climbing upstairs, climbing downstairs, standing, and jumping. Six features were extracted from the data namely mean, correlation, energy, variance, skewness, and signal magnitude area. The linear and nonlinear models implemented were linear kernel and Radial basis function kernel. The best results were obtained by a grid search method. Different combinations of features were used to build the model and its performance was analyzed. The features that gave the highest precision of 95 % were signal magnitude area, energy, correlation, and skewness.

3. Prediction of Insufficient Accuracy for Human Activity Recognition Using Convolutional Neural Network in Compared with Support Vector Machine

M. S. Saravanan and S. Charan are conducting a detailed study of the two machine-learning algorithms to understand their merits and demerits in predicting human activities [3]. The two MLCs studied in this project are the Support Vector Machine and the novel Convolutional Neural Network. Forty-seven samples were collected for the analysis. A 70:30 train test split ratio was used. The accuracy obtained from both models was 90.19 % for SVM and 92.46 % for CNN. The paper states CNN is a better model for Human Activity Recognition.

4. Comparison of Machine Learning and Deep Learning Approaches for Human Activity Recognition

In [4] several Machine Learning and Deep learning algorithms are put to the test to understand their performance in Human Activity Recognition. The models considered in this study are CNNLSTM, Logistic Regression, Bi-Directional LSTM, SVM with RBF kernel, LSTM, and CNN. The results show the accuracy of each model and the time taken for training in each model. The accuracy for LSTM, CNN, CNN-LSTM, Bi-Directional LSTM, SVM, and Logistic Regression is 95%, 91%, 97%, 97%, 89%, and 90% respectively. The time of training (in minutes) for the models LSTM, CNN, CNN-LSTM, Bi-Directional LSTM, SVM, and Logistic Regression are 322, 245, 957, 575, 78, and 93 respectively.

5. Audio Classification Method Based on Machine Learning

The study conducted in [5] to develop an audio classifier gives a detailed insight into the different structures of audio signals such as audio frame, clip, shot, and high-level semantic unit. The paper also extracts three audio features namely Mel-Frequency Cepstral Coefficients, Zero Crossing Rate, and Short Time Energy. The ML classifier used SVM (Gaussian kernel). The two databases used to evaluate the performance are general sounds and audio scenes which results in an accuracy of 87% and 86% respectively.

LIFE CYCLE STAGES OF THE PROJECT

The project adopted a Waterfall model of Software Development Life Cycle (SDLC). It is widely popular and one of the oldest SDLC models. In this model the processes are sequential and only after the 100% completion of one step does the model move to the next step.

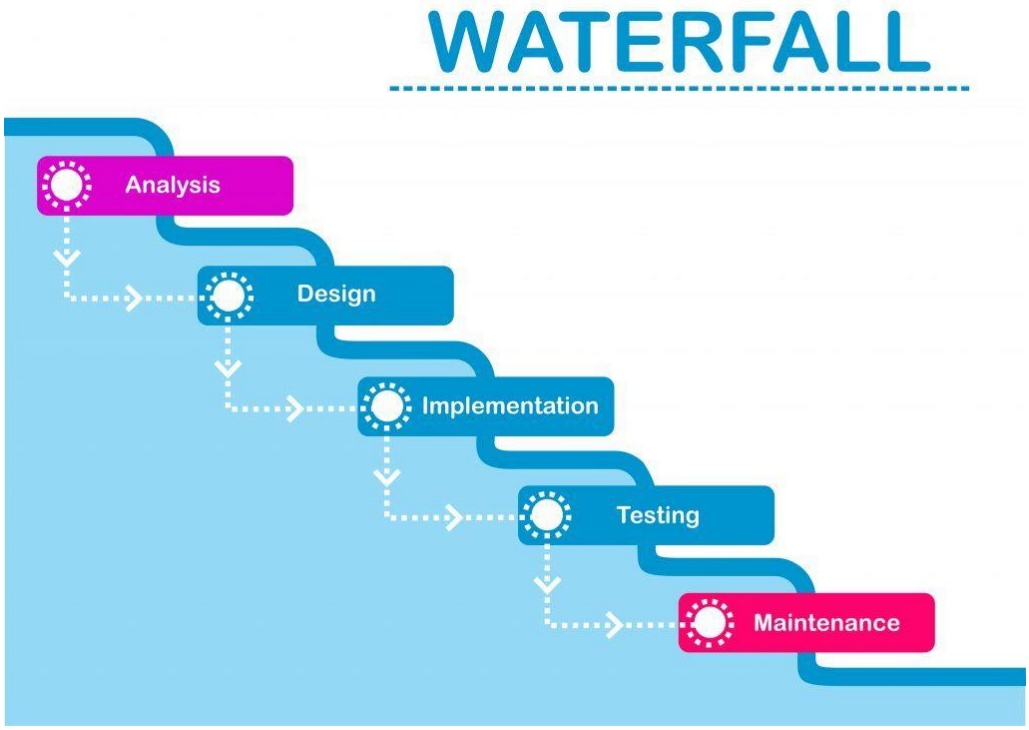


Figure 1 Waterfall Methodology [8]

Agile methodology is best suited in current software development projects where multiple teams are working together and frequent interactions with the client are needed and regular updates are made on the end product. This project is an MSc masters project done individually under supervision of an assigned professor. The goal of the project is set in the beginning of the semester and followed by which there is very little to no changes. Also, the project is an individual task, there are no multiple people working on

the same project at the same time. All these are the reasons for choosing a Waterfall model over Agile model for development of the project. The development has been three step sequential process.

1. Software development: the software part of the project was developed first, which included data collection and model training for an audio classifier. Python language and PyCharm IDE [10]

2. Hardware development: followed by the development of the audio classifier, the focus was on the setup of the hardware in this project. Raspberry Pi and a USB microphone are the two hardware components used for capturing audio signals.

3. Documentation: the final step of the project was the documentation where the project has been documented in a 7-page report of approximately 5,000 words along with which other supporting documents were also prepared.

Validation and Verification in SDLC

The whole SDLC was supervised by the designated professor. Weekly meetings were conducted with the supervisor for discussion on the project. Outcomes obtained and challenges faced were discussed in detail every week, feedback and possible approaches were talked through on the same. The SDLC was tracked using a Gantt chart.

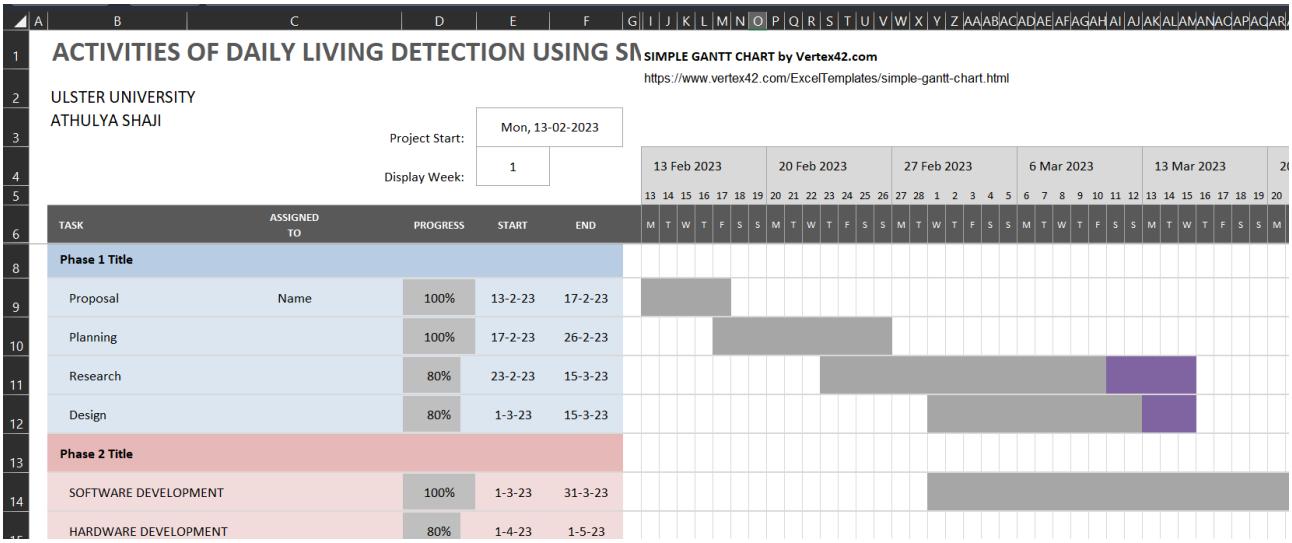


Figure 2: Gantt Chart

DEVELOPMENT OF THE PROJECT

Data Collection

The data for training the model for detection of Activities of Daily Living (ADLs) were from the open source website Pixabay (https://pixabay.com/sound-effects/search/flush/?theme=household) [9]. The five classes of audios used in this project are door opening and closing sounds, running water from a faucet, hair dryer, microwave, and washing machine. Around 40 samples of each set were downloaded from Pixabay.

The two sets of databases used in this project are shown in the below table.

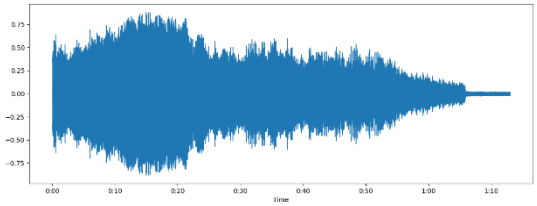
Table 1: Datasets

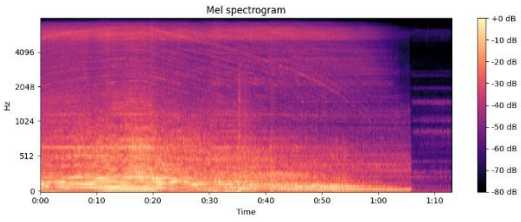
Classes	Count
Door	20
Faucet	20
Hair dryer	20
Microwave	20
Washing machine	20

Classes	Count
Door	40
Faucet	40
Microwave	38
Washing machine	39

Data Preprocessing

The data downloaded from the website was in MP3 format which were converted into WAV format, along with which the metadata of each audio file was created and stored in a CSV file as shown in Figure 2. The audio files were later converted into a set of data arrays with a sample rate of 22050. Which was again used to convert the audio signals to a spectrogram. The Mel Frequency Cepstral Coefficients (MFCCs) were derived from each of these audio signals. The Librosa library was used for all the preprocessing [11].





	A	B	C	D
1	slice_file_name	fold	classID	class
2	071788_door-openclose_1-2aif-89957.wav	1	1	Door
3	AnyConv.com__door-opening-and-closing-46085.wa	1	1	Door
4	AnyConv.com__front-door-open-and-close-89835.w	1	1	Door
5	AnyConv.com__front-door-open-close-lock-28832.v	1	1	Door
6	AnyConv.com__modern-front-door-inside-and-outsi	1	1	Door
7	AnyConv.com__opening-and-closing-creaky-door-30	1	1	Door
8	AnyConv.com__opening-closing-and-locking-door-43	1	1	Door
9	AnyConv.com__opening-the-front-door-47333.wav	1	1	Door
10	AnyConv.com__room-door-opening-and-closing-279	1	1	Door
11	AnyConv.com__unlocking_opening-front-door-4446	1	1	Door
12	AnyConv.com__upvc_door_opening_and_closing-50	1	1	Door
13	AnyConv.com__wood-door-open-and-close-26650.v	1	1	Door
14	closing-door-101360.wav	1	1	Door
15	door-close-48121.wav	1	1	Door
16	door-closing-46357.wav	1	1	Door
17	door-closing-door-closed-6006.wav	1	1	Door
18	door-lock-43124.wav	1	1	Door
19	door-open-45407.wav	1	1	Door

Figure 2 Audio visualization and CSV file of Metadata

Audio Classification

Artificial Neural Network (ANN), Naïve Bayes (NB), Support Vector Machine (SVM), k-Nearest Neighbors (kNN), Decision Tree, and Perceptron are the six classifying algorithms used to train the audio classifier.

Few outputs from the training the MLCs on a 20 sample 5 class dataset with a split ratio of 80:20 are as follows.

```
Epoch 100/100
1/3 [=========>....................] - ETA: 0s - loss: 1.6472 - accuracy: 0.4688
Epoch 100: val_loss did not improve from 0.96636
3/3 [==============================] - 0s 13ms/step - loss: 1.9774 - accuracy: 0.4211 - val_loss: 1.2187 - val_accuracy: 0.5000
Training completed in time:  0:00:03.887506
0.5
```


```
Epoch 100/100
1/3 [=========>....................] - ETA: 0s - loss: 1.6472 - accuracy: 0.4688
Epoch 100: val_loss did not improve from 0.96636
3/3 [==============================] - 0s 13ms/step - loss: 1.9774 - accuracy: 0.4211 - val_loss: 1.2187 - val_accuracy: 0.5000
Training completed in time:  0:00:03.887506
0.5
```


Figure 3: ANN with MFCC 40 AND MFCC 20

```
Accuracy: 0.7
Cross-validation scores:  [0.625      0.5        0.625      0.625      0.875      0.75
 0.71428571 0.71428571 0.57142857 0.42857143]
Mean CV score:  0.6428571428571429
Precision:  0.7
Recall:  0.7
```

```
Accuracy: 0.7
Cross-validation scores:  [0.625      0.5        0.625      0.625      0.875      0.75
 0.71428571 0.71428571 0.57142857 0.42857143]
Mean CV score:  0.6428571428571429
Precision:  0.7
Recall:  0.7
```

Figure 4: KNN with MFCC 40 AND MFCC 20

Few outputs from the training the MLCs on a 40 sample 4 class dataset with a split ratio of 80:20 are as follows.

```
Epoch 100/100
1/4 [======>.....................] - ETA: 0s - loss: 1.7305 - accuracy: 0.4375
Epoch 100: val_loss did not improve from 0.99900
4/4 [==============================] - 0s 13ms/step - loss: 1.6358 - accuracy: 0.4365 - val_loss: 1.0450 - val_accuracy: 0.4375
Training completed in time:  0:00:05.660150
0.4375
```

```
Epoch 100/100
1/4 [======>.....................] - ETA: 0s - loss: 1.3865 - accuracy: 0.3125
Epoch 100: val_loss did not improve from 0.93125
4/4 [==============================] - 0s 11ms/step - loss: 1.7261 - accuracy: 0.3810 - val_loss: 1.2192 - val_accuracy: 0.5000
Training completed in time:  0:00:05.922499
0.5
```

Figure 5: ANN with MFCC 40 AND MFCC 20

```
Accuracy: 0.6875
Cross-validation scores:  [0.69230769 0.53846154 0.69230769 0.53846154 0.69230769 0.69230769
 0.58333333 0.5        0.75       0.66666667]
Mean CV score:  0.6346153846153847
Precision:  0.6875
Recall:  0.6875
```

```
Accuracy: 0.6875
Cross-validation scores:  [0.69230769 0.53846154 0.69230769 0.53846154 0.61538462 0.69230769
 0.58333333 0.5        0.75       0.66666667]
Mean CV score:  0.626923076923077
Precision:  0.6875
Recall:  0.6875
```

Figure 6: KNN with MFCC 40 AND MFCC 20

```
Accuracy: 0.65625
Cross-validation scores:  [0.61538462 0.69230769 0.61538462 0.46153846 0.69230769 0.61538462
 0.83333333 0.58333333 0.83333333 0.66666667]
Mean CV score:  0.660897435897436
Precision:  0.65625
Recall:  0.65625
```

```
Accuracy: 0.625
Cross-validation scores:  [0.76923077 0.61538462 0.46153846 0.38461538 0.76923077 0.46153846
 0.91666667 0.41666667 0.66666667 0.5       ]
Mean CV score:  0.5961538461538461
Precision:  0.625
Recall:  0.625
```

Figure 7: Perceptron with MFCC 40 AND MFCC 20

The result of training SVM model on a 20 sample 5 class dataset with a train-test ratio of 70:30 are as follows:

```
Cross-validation scores:  [0.85714286 0.71428571 1.         0.85714286 0.57142857 1.
 0.71428571 0.66666667 1.         0.5       ]
Mean CV score:  0.7880952380952382
Test score:  0.7931034482758621
Accuracy: 0.79
Precision:  0.7931034482758621
Recall:  0.7931034482758621
Cross-validation scores:  [0.57142857 0.71428571 0.85714286 0.85714286 0.42857143 1.
 0.71428571 0.5        1.         0.5       ]
Mean CV score:  0.7142857142857143
Test score:  0.7931034482758621
Accuracy: 0.79
Precision:  0.7931034482758621
Recall:  0.7931034482758621
```

Figure 8: SVM with MFCC 40 AND MFCC 20

The SVM model trained on 20 sample 5 class dataset with MFCC 40 and a train-test ratio of 70:30 was selected as the audio classifier in this project.

Raspberry Pi and USB microphone

Raspberry Pi 3 model B+ along with USB microphone was used to capture ADLs in real time and transferred to the computer.

Figure 9: Raspberry Pi and USB microphone [6] [7]

The USB microphone was integrated with Raspberry Pi and the record command was used to capture ADL sound. The audio signal captured is stored in the home directory of the Raspberry Pi which is then transferred to the system by connecting Raspberry Pi and Laptop using a SSH client Putty [12].

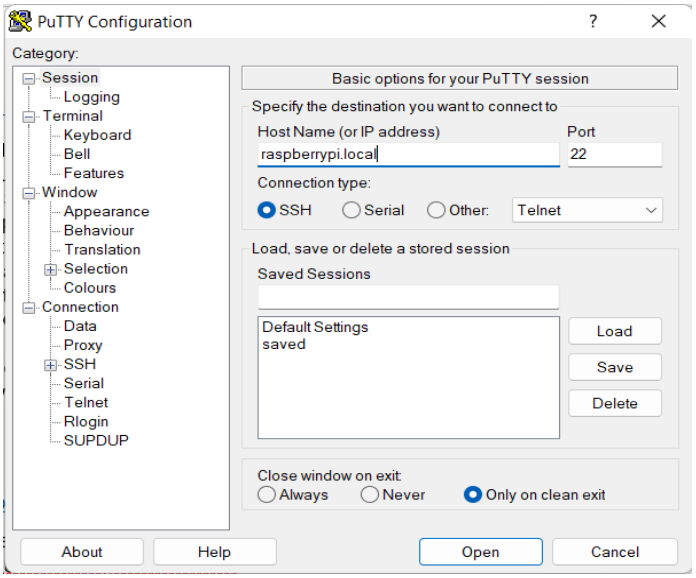


Figure 9: Putty Client

Audio Detection

The ADL sound received from the Raspberry Pi is fed to the audio classifier to detect the sounds. If the audio signal is predicted with an accuracy greater than or equal to 70 % then the log of the activity is stored in a CSV file as shown in the Figure 12
The code snippet shows the prediction of an audio signal as door with an accuracy of 80% using the SVM audio classifier.

```
# Load the saved SVM model from the file
svm = joblib.load('svm_model.pkl')
# load the audio file
filename="C:/Users/DELL/Desktop/MP/Dataset_wav_20/fold1/door-opening-and-closing-6232.wav"
# Extracting MFCC's of the audio file
audio, sample_rate = librosa.load(filename, res_type='kaiser_fast')
mfccs_features = librosa.feature.mfcc(y=audio, sr=sample_rate, n_mfcc=20)
mfccs_scaled_features = np.mean(mfccs_features.T,axis=0)
mfccs_scaled_features=mfccs_scaled_features.reshape(1,-1)
# predition using the SVM
predicted_label=svm.predict(mfccs_scaled_features)
pred_proba = svm.predict_proba(mfccs_scaled_features)
print(predicted_label)
print(pred_proba.max())
```

Figure 10: Code snippet

```
['Door']
0.8077069171821358
```

Figure 11: Output from SVM

	A	B	C
1	Date	Time	Activity
2	10-05-2023	13:43:46	['Door']
3	10-05-2023	14:05:50	['Door']
4	10-05-2023	15:15:30	['Faucet']
5	10-05-2023	16:30:46	['Faucet']
6	10-05-2023	17:10:30	['Washing Machine']
7	13-05-2023	15:30:26	['Door']
8			

Figure 12: CSV file of ADL Log

The process of semi-automated annotation has been detailed here. If the audio signal is predicted with an accuracy less than 70 %, a user input is taken for the identification of the unidentified audio signal. In this case if the user input is an existing class of the dataset, the audio signal will be added to the respective folder or else a new class will be created, and the audio signal will be stored in it. The metadata is also created and added into the CSV file (Figure 2). The figure below shows an output from the classifier with accuracy 67%. The user input is asked for identifying the audio signal. In this example a new class called Dryer was created. The dataset will be updated as shown in Table 2. Figure 14 shows the metadata and ADL log created in the current scenario.

```
['Washing Machine']
0.6741585904476471
Is it a sound of ['Washing Machine']  : yes/no
no
What sound is it !!
Dryer
```

Figure 13: Output from SVM

Table 2: Updated Dataset

Classes	Count
Door	20
Faucet	20
Hair dryer	20
Microwave	20
Washing machine	20
Dryer	1

	A	B	C	D
1	slice_file_nam	fold	classID	class
98	audio20	5	5	Washing Machine
99	audio21	5	5	Washing Machine
100	audio22	5	5	Washing Machine
101	audio1	6	6	Dryer
102	audio2	6	6	Dryer
103	audio1	7	7	Fan

	A	B	C
1	Date	Time	Activity
23	15-05-2023	01:36:58	Washing Machine
24	15-05-2023	01:37:26	Washing Machine
25	15-05-2023	01:45:26	Washing Machine
26	15-05-2023	01:45:42	Dryer
27	15-05-2023	01:47:01	Dryer
28	15-05-2023	01:50:38	Fan

Figure 14: Updated CSV files of Metadata and ADL Log

Challenges During the Project Development

The major challenge faced during the development of this project was:

1. Availability of dataset for the required classification in the project was scarce. The MP3 files used from Pixabay were limited in numbers. Hence deep learning algorithms gave poor performance.
2. During the initial setup of Raspberry Pi to install the Raspbian OS, the university computer lab monitors where used. The campus WI-FI (eduroam) network was not compatible with the Raspberry Pi. Different trouble shooting methods such as reconfigured the WPA_ supplicant.conf, changing the Raspbian OS version. However, the process was not successful. Hence personal WI-FI was used while working on Raspberry PI
3. An approach to develop the audio classifier on the Raspberry Pi itself was considered. There were many challenges for it. Python library Librosa, significant for feature extraction from audio file was throwing error while installation on Raspberry Pi. Different Raspbian OS versions were tried but it was difficult to get all the library needed for the audio classifier to be installed on Raspberry Pi. Therefore, the audio classifier was developed on the PC.
4. Another challenge was the integration of MATRIX Voice with Raspberry Pi to capture sounds. According to the initial proposal a MATRIX Voice was planned to capture ADL sounds in real time. But the integration of both the devices gave an error. Different configuration processes as per the MATRIX.ONE documents were tried as part of the troubleshooting. Unfortunately, it was unsuccessful. Therefore, a USB microphone was used for capturing ADL sounds in real time.

Additional Documents

Additional documents attached along with this supporting document are:

1. Source code
   The folder contains nine python code files as follows:
   1. Data visualization: is to represent the audio signals as wave ans spectrogram

2. data preprocessing: is to convert MP3 file to WAV and create metadata
3. ANN Model: ANN classifier implemented
4. KNN Model: KNN classifier implemented
5. NB Model: NB classifier implemented
6. SVM Model: SVM classifier implemented
7. Decisiontree Model: Decision tree classifier implemented
8. Perceptron Model: Perceptron classifier implemented
9. Audio Classifier: ADL detection using selected model and semi-automated annotation method for unidentified ADL sounds

2. Dataset
   There are two folders, one with 4 classes and approximately 40 samples each and another with 5 classes and 20 samples each along with two more classes created as part of identifying new audio signals.
3. Metadata of dataset
4. ADL log file
5. Gantt Chart

REFERENCES

[1] A. Palaniappan, R. Bhargavi and V. Vaidehi, "Abnormal human activity recognition using SVM based approach," 2012 International Conference on Recent Trends in Information Technology, Chennai, India, 2012, pp. 97-102, doi: 10.1109/ICRTIT.2012.6206829.

[2] Z. A. Sunkad and Soujanya, "Feature Selection and Hyperparameter Optimization of SVM for Human Activity Recognition," 2016 3rd International Conference on Soft Computing & Machine Intelligence (ISCMI), Dubai, United Arab Emirates, 2016, pp. 104-109, doi: 10.1109/ISCMI.2016.30.

[3] M. S. Saravanan and S. Charan, "Prediction of Insufficient Accuracy for Human Activity Recognition using Convolutional Neural Network in Compared with Support Vector Machine," 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), Uttar Pradesh, India, 2022, pp. 1915-1919, doi: 10.1109/IC3I56241.2022.10072905.

[4] B. Moradi, M. Aghapour and A. Shirbandi, "Compare of Machine Learning and Deep Learning Approaches for Human Activity Recognition," 2022 30th International Conference on Electrical Engineering (ICEE), Tehran, Iran, Islamic Republic of, 2022, pp. 592-596, doi: 10.1109/ICEE55646.2022.9827335.

[5] F. Rong, "Audio Classification Method Based on Machine Learning," 2016 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Changsha, China, 2016, pp. 81-84, doi: 10.1109/ICITBS.2016.98.

[6] (No date b) Prismic.io. Available at: https://images.prismic.io/rpf-products/bef8cda3-64ea-4098-

bf188e731a6e9a0d_3b%2B%20Angle%202.jpg?ixlib=gatsbyFP&auto=compress%2Cformat&fit=max&w=799&h=533 (Accessed: 11 May 2023).

[7] (No date d) Shopify.com. Available at: https://cdn.shopify.com/s/files/1/0176/3274/products/mini-usb-microphone-the-pi-hut-10286533678856618179_grande.jpg?v=1646920627 (Accessed: 11 May 2023).

[8] August, M. (2022) Waterfall product development, Agile First. Available at: https://agilefirst.io/waterfall-product-development/ (Accessed: 14 May 2023).

[9](No date) Pixabay.com. Available at: https://pixabay.com/sound-effects/search/flush/?theme=household (Accessed: 11 May 2023).

[10] Download PyCharm: Python IDE for professional developers by (no date) JetBrains. Available at: https://www.jetbrains.com/pycharm/download/ (Accessed: 11 May 2023).

[11] McFee, B. (2019) Why resample on load?, librosa blog. Available at: https://librosa.org/blog/2019/07/17/resample-on-load/ (Accessed: 12 May 2023).

[12] Download PuTTY - a free SSH and telnet client for Windows (no date) Putty.org. Available at: https://www.putty.org/ (Accessed: 12 May 2023).