Project Title:
Supported Hearing in Elderly Citizens

Project acronym:
SHiEC

Contract No:
AAL-2013-6-065

Deliverable Reference: D1.1 Algorithm to detect own voice	Date: 26/10/2015

Title:

Own voice detection for linguistic datalogging

May 2014 – Nov 2015

Author: Obaid ur Rehman Qazi

Contributors: /

Type of Document: Restricted

Contents

1. INTRODUCTION

The impact of hearing loss can be categorized as functional, social, emotional and economic. Functionally, the individual loses the ability to communicate fluently with others. Since almost all occupations, be it professional or non-profit, require interaction through spoken language, hearing loss has profound social, emotional and economic consequences. Limited access to services and exclusion from communication can have a significant impact on everyday life, causing feelings of loneliness, isolation and frustration, particularly among older people and a recent study found a convincing link between hearing loss and onset of dementia[1]. There is more and more evidence showing that treatment of hearing can have a positive impact on a person's general health situation[2].

Hearing implants are relatively complex systems. They consist of an internal implanted part and an external processor, with batteries, charging devices, cables, coils and further accessories to improve connectivity, e.g. to listen to the TV. To further improve hearing outcomes and sound quality, manufacturers have recently introduced many new signal cleaning algorithms such as noise reduction systems, dual microphone beamforming, wind noise reduction & impulse noise reduction. This means that the newer devices support multiple programs for optimal hearing in different use cases, e.g. hearing in a restaurant or public place, enjoying music or using the phone. However, in the past it was the recipient who had to manually choose the optimal program for a given environment and ensure that it is correctly maintained. This disadvantages the older person, particularly those with minimal ICT skills implying that the benefits they could obtain from these powerful devices are rarely realized.

Recent advances in the field, which include data logging, mean that more automated systems could be developed focusing primarily on the needs of the older person. Data logging can be used to monitor the usage of the hearing implant system and monitor the listening environment of the elderly users. If a device is no longer optimally functioning, events are stored in the memory and auditory warnings given to the recipient, indicating that the device has to be serviced. Data logs also store information about the user profile, including how many hours a day the device is used and the type of environment e.g. quiet, noisy, rich in terms of spoken language, music etc. This feature enables elderly person & their family/care givers feel they are achieving their goal to participate in spoken language conversations and the hearing world.

Clinicians (audiologists, speech therapists …) also value the insights in how customers use their devices and in which acoustic environment they function[3]. These data logs enable them to refine their counselling activities based on a better understanding of user needs and to provide more focused training for a particular recipient. This data also permit them to better analyse/trouble shoot the technical issues with the device. Furthermore this data helps to fine tune the device parameters based on the user needs and hearing outcomes.

[1] F. Lin et al, Hearing loss and cognition in the Baltimore Longitudinal Study of Aging, Neuropsychology, 2011

[2] J. Nachtegaal, Hearing ability in working life and its relationship with sick leave and self-reported work productivity, Ear Hear, 2012.

[3] DiaLog™: A clinician's guide to data logging, http://www.cochlear.com/wps/wcm/connect/uk/for-professionals/rehabilitation-resources/dialog , last accessed 30/10/2015.

1.1 1st generation data-logging in Nucleus 6

Self-reporting is subjective and not always accurate. If a patient is not progressing as they should, use of data logging can help them reach their goals sooner. Data logging is a feature in the Cochlear™ Nucleus ® 6 System that provides with valuable, evidence-based feedback about patient's listening environment and device usage. Accurate insights can help to troubleshoot faster, deliver more effective counselling and save precious clinical time in the longer term.

Whenever sound processor is switched on, information is stored related to device use, sound environment, exposure to speech and accessory use. When user's returns to the clinic, this information is uploaded from the sound processor by Custom Sound Software® and displayed as 24-hour averages since the last time the data was analyzed. Time- and event-based data is recorded every five minutes. This data is stored for up to 32 years, so if a patient misses a clinic visit; it will still be available to download.

With data logging as part of a holistic approach, enables us to quickly and easily identify areas of need and possible barriers to progress. This should enable immediate and targeted troubleshooting and counselling to promote lasting change and improved outcomes. Data logging information also helps patients get the most out of their device and improve outcomes.

1.2 What data is recorded

The data-logging module records the device usage (Time on Air), specific events (Coil Off, Power On), Environmental Scenes, Program usage, Accessory Usage, and Volume and Sensitivity [see DialLog for more details]. The Environmental Classifier analyses the incoming sound and classifies the auditory situation (scene) as one of 'Speech', 'Speech in Noise', 'Noise', 'Music', 'Wind noise' or 'Quiet' classes. It also logs the level of the environmental sounds expressed in dBA. An example scenes categorization of the everyday listening environments experienced by an implant user is shown in figure 1. This scene graph reports on the average time each day a patient spent in a particular listening environment. The classifier engine in the processor automatically samples the sound environment and assigns it to one of the six categories as mentioned above.

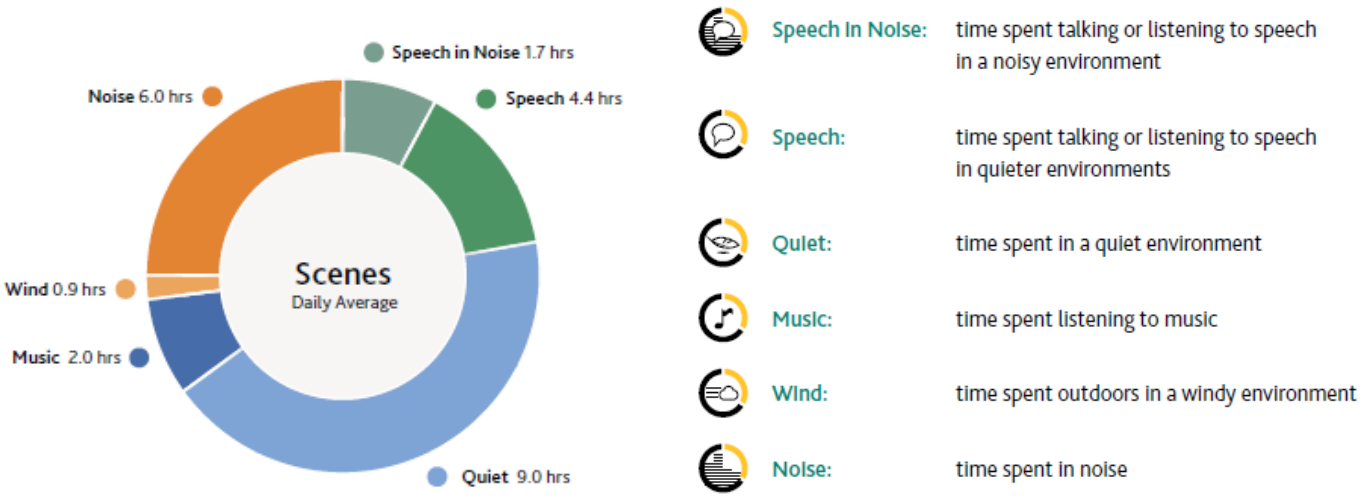

Figure 1: An example scenes categorization of the everyday listening environments experienced by an implant user displayed as 24-hour averages.

1.3 Missing 'Own Speech' Class

Although 1st generation data logging does discriminate between speech in quiet and speech in background noise. However it is not discriminating between speaker's own speech and the partners' speech. This is unfortunate, as the amount of own speech is a good indicator of social participation and for children language development. The amount of caregiver's speech and the number of conversational turns reflect the quality and adequacy of the language input for children. Both measures together can provide insight about the social integration and participation of elderly citizens. These measures can also equip clinicians with important information to guide their therapy and help researchers to increase knowledge about the everyday experience of people with cochlear implants. To increase the scientific and clinical value of automated auditory scene classification within the cochlear implant sound processor, the detection of the wearer's own speech will be crucial. It is the basic requirement for analysing several important features of the linguistic experience. To fulfil this gap the development of own voice algorithm was included in work package 1 as task 1.1 of the SHiEC project. The deliverable for this project is a own voice detection algorithm.

This document is intended to provide some higher level details as to how the current design (OVAD version V1.0) was arrived at, and the tools that were used in its development. Its main purpose is to document the evolution and evaluation of the algorithm to foster an informed decision about the maturity of this feature for its implementation in firmware for further chronic evaluation.

2. PROJECT PLAN

The project was subdivided into 3 phases. Currently the project is in the Phase 3 stage. The deliverable of the WP1.1 is an al

A short description of each phase is given below.

Phase 1: Feasibility [May 2014 – Dec 2014]

Phase 2: Optimized Design, implementation and Evaluation [Jan 2015 – June 2015]

Phase 3: Real-time platform implementation [July 2015- Nov 2015]

2.1 Phase 1: Feasibility

In phase 1 of the project different design options were explored and the design requirements were set. In this phase a reasonably large database of own/external speech was also recorded and then different features were explored which could be used in the final design of the classifier.

2.1.1 Design Goals and Requirements: Own Voice Activity Detection

The first design proposal of the 'Own Speech Classifier' is solely intended for the data logging purposes and is not envisioned for driving the sound processing signal path at this stage. Furthermore, the technological approach is maximally aligned with the SCAN environmental classifier[4] where a number of non-adaptive features are calculated in a straight forward fashion from the SP16 signal path and a decision tree is used to classify the incoming speech into 'own' or 'external' speech. The ideas is to design and evaluate a simple own voice detection algorithm using maximally the available SCAN classifier design to expedite the development process. However, the

[4] DiaLog™: A clinician's guide to data logging, http://www.cochlear.com/wps/wcm/connect/uk/for-professionals/rehabilitation-resources/dialog , last accessed 30/10/2015.

own voice activity detection has to be good enough to detect trend across time. We may accept up to 10% systematic bias. Ideally the accuracy should be above 80% in all the cases which is comparable to the LENA device which processes the speech offline. The minimum accuracy requirements on the training and tests sets are given below.

- Own voice activity detection has to be good enough to detect trend across time. Up to 10% bias is allowed if systematic.
- Accuracy of the own voice time in a child – adult conversation > 80% in a conversation in good auditory scene (SNR > 15 dB).
- Accuracy of the own voice time in an adult – adult (same gender) conversation > 80% in a conversation in good auditory scene (SNR > 15 dB).
- Accuracy in turn taking in a child – adult conversation > 80% in a conversation in good auditory scene (SNR > 15 dB).
- Accuracy in turn taking in a adult – adult (same gender) conversation > 80% in a conversation in good auditory scene (SNR > 15 dB).

2.1.2 Design: Own Voice Activity Detection (OVAD)

After exploring different options it was decided to build the OVAD algorithm upon the existing Environmental Classifier implementation which analyses the incoming sound and classifies the auditory situation as one of 'Speech', 'Speech in Noise', 'Noise', 'Music', 'Wind' or 'Quiet' class. The OVAD uses some of SCAN features plus two additional features to discriminate 'own speech' from 'external speech' (see figure 2). Compared to the SCAN which smoothly switches the user preferred settings depending on the environment as specified in Custom Sound the OVAD algorithm on the other hand does not alter any program settings but is used to log an additional class named as 'Own Speech'. Furthermore, classifier counts the number of conversational turns taken during the everyday conversations and therefore an estimate of utterance duration can be inferred (figure 3).

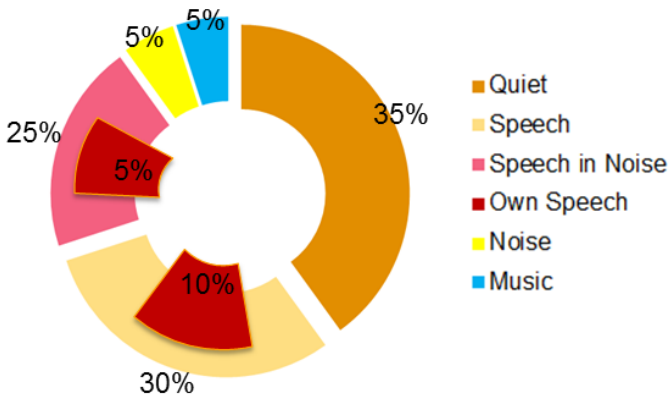

Figure 2: An example of the Classification where speech is further classified as 'Own' and 'External' speech.

[5] CP900 Environmental Classifier Design Description [358375].

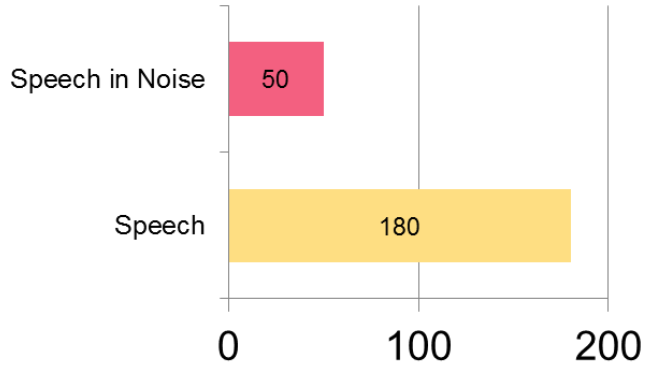

Figure 3: An example of the Classification where the number of conversation turns in 'Speech' and Speech in Noise' are logged.

A block diagram is provided in Figure 4 which gives an overview of the classifier. It is implemented in a manner typical of audio classification schemes: a series of features are extracted from the incoming audio signal and those are passed into a classification block which determines the most likely class based on those features. The class decision is then smoothed to prevent class transitions occurring too frequently, as well as to increase the level of certainty that the determined class is correct.

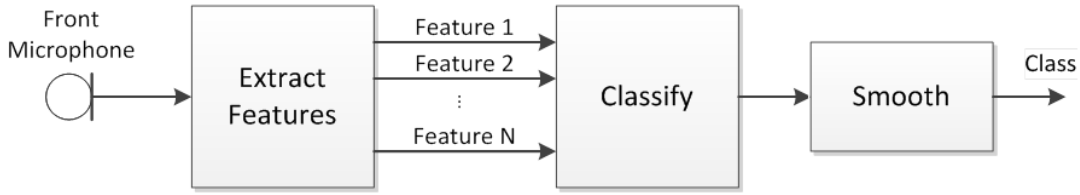

Figure 4: Block diagram of the Own Voice Activity Detection (OVAD) Classifier

2.1.3 Sound Database

In order to design and train a classifier like this, it is really important to have a good database of sound samples which have been manually classified, and which provide good coverage of the range of different sounds you would expect to find in each class during real world usage. The recordings were done with SP15[5] microphone shells for the 'own' and 'external' speech. The database contains 7 male and 5 female speakers out of which four were children. The whole database contains around 3 hours of speech recordings out of which half is recorded as 'Own Speech' when the microphone shell was placed on the speaker's own hear. The remaining half is recorded as 'External Speech' when the microphone shell was placed on a manikin and the speaker spoke from a distance. Thus we have more than 1 hour of labelled speech for 'Own' and 'External' class. The recordings were done only in the office and home environment in relatively low noise conditions. Additional noise was also added to simulate the noisy conditions (15, 10 and 5 dB SNR). Apart from these recordings some test conversations recordings were also done to evaluate a conversation scenario where two speakers are talking in a semi-realistic manner. The database used to develop and train this classifier is stored on Peforce.

[6] SP15 DSP Firmware Functional Design Description [238212].

2.1.4 Design Approach

The design of the OVAD is based on the existing Environmental classifier which uses 5 features and a decision tree to classify the input signal into 5 different classes. The OVAD utilizes the available features of Environmental Classifier plus some additional features and a new decision tree is trained on the own and external speech recordings in order to classify the speech as 'own' or 'external' speech. The flow chart in figure 5 shows how the Environmental Classifier and OVAD Classifier are integrated together in a Simulink model. The OVAD classifier uses the Environmental Classifier features plus some additional features which are extracted from the input audio signal and are provided to the OVAD decision tree and the class is selected. In case the Environmental Classifier is in 'Speech' or 'Speech in Noise' environment the OVAD outputs the class as 'Speech', 'Speech in Noise', 'Own Speech' or 'Own Speech in Noise'. To reduce the complexity in order to implement this algorithm on to SP16 it was decided to add minimum number of additional features to achieve the design requirements.

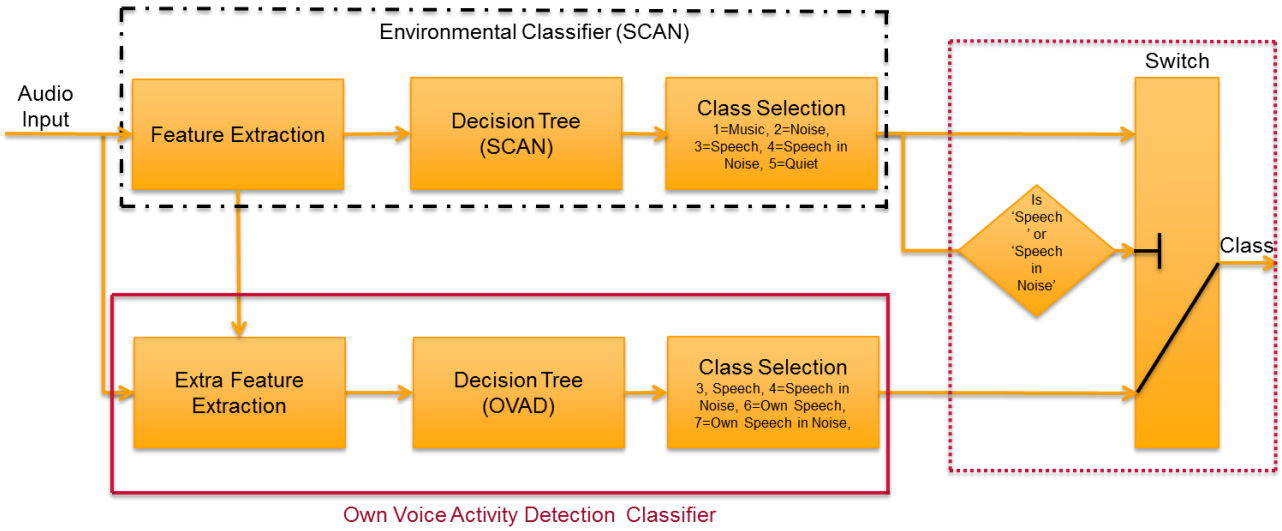

Figure 5: Block diagram of the Own Voice Activity Detection (OVAD) Classifier in combination with the Environmental classifier.

2.1.5 Features Exploration

All the features used in the Environmental classifier plus some additional features were selected as the starting point for the 'Own Voice Activity Detection' and the separation power of different combinations was evaluated. These features use the different characteristics of the near filed and far field signals to segregate 'Own Speech' form the 'External Speech'.

2.1.6 Results

The classification results based on all the tested features are shown in figure 6 for different update rates on the clean recording database. For both training and test data the accuracy is above 90% for all the cases. Here the accuracy is defined as the percentage of segments which were correctly identified as own and external speech. The accuracy is slightly higher for the faster update rates. Therefore it is desirable to have higher update rate of 100 ms to detect short 'yes'/'no' kind of responses.

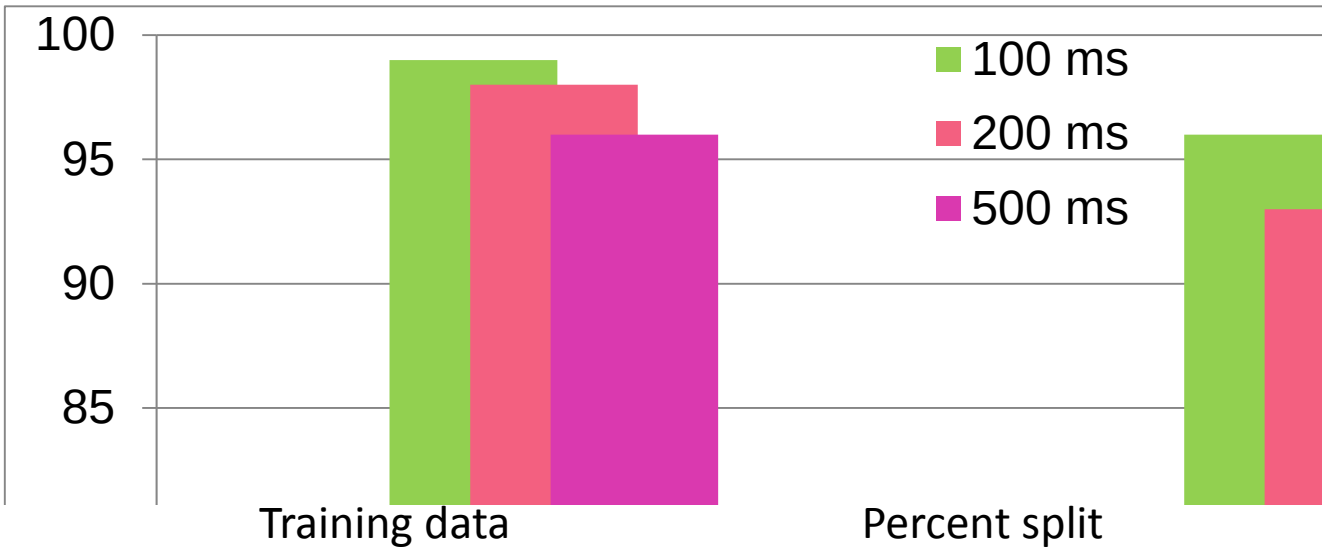

Figure 6: Classification accuracy for different update rates on training and split test data using all the 7 features.

2.2 Phase 2: Optimized Design, Implementation and Evaluation

Based on the feasibility studies and different design options an optimized design where we concentrated on a solution which could fit within the computational boundaries of the sound processor was achieved.

2.2.1 Architecture

For the implementation the samples are taken from the audio file which is recorded by the SP15 microphone shells but only the output of the front omni microphone is used. As shown in Figure 7 below, once the new samples are received different features are calculated. Note that three of the features work on the Log Power Spectrum of the input so this Log Spectrum is calculated only once. The calculated features are then input to the decision tree, which classifies the current set of features. The features are calculated much more regularly, but their outputs are smoothed prior to being sampled by the decision tree. The output class is then smoothed and made available for the logging purposes. Thee time domain and 3 frequency domain features are finally selected which contribute most to the accuracy of the classification and therefore are recommended for the final design.

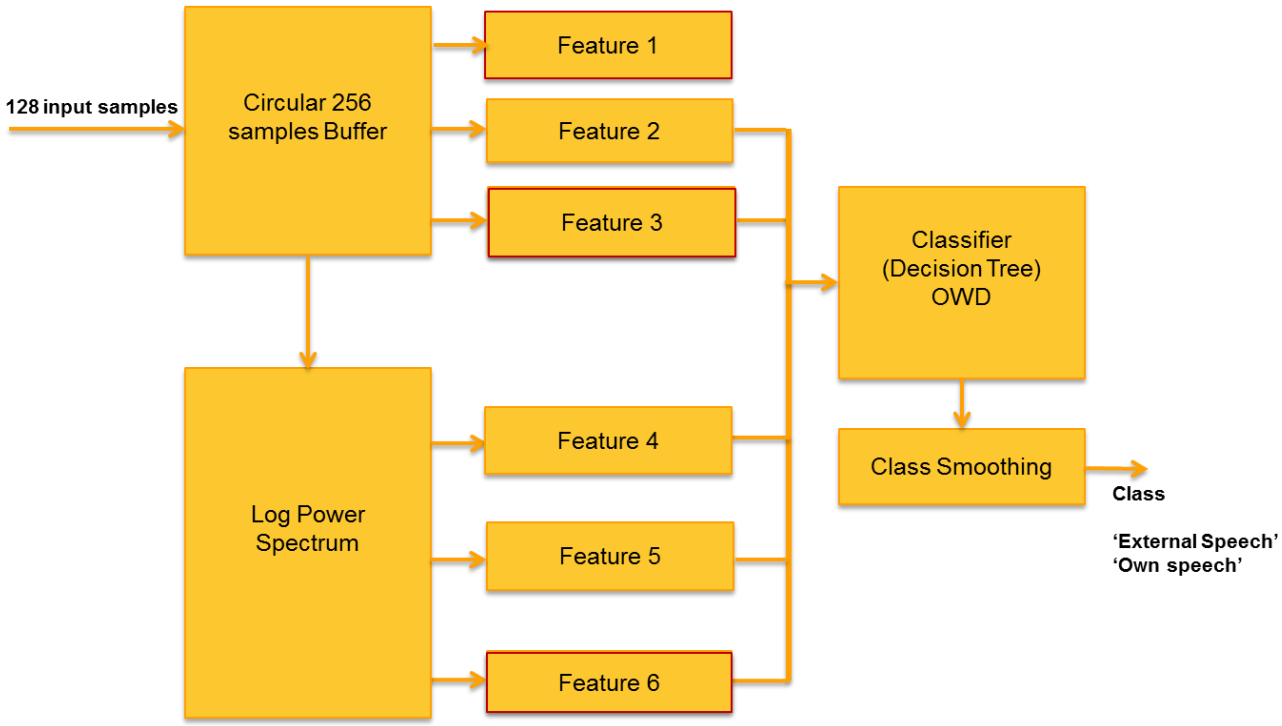

Figure 7: High level architecture of the OVAD Classifier. Two new features are added to the classifier and provided to the decision tree (shown in red outline). The decision tree is trained on the new set of recorded database containing 'Own' and 'External' speech recordings.

2.2.2 Implementation

Most of the design and the initial implementation of the environmental classifier was done in Simulink. A model containing the Simulink implementation of the classifier is stored in Perforce. This model can be run in real time on the xPC machine. The Real-Time Target Machine usually called the xPC machine is a ready-to-use, high-performance, real-time simulation and testing platform with built-in analog and digital I/O channels. It includes all required cables, terminal boards, and adapters to enable easy connectivity from the target machine I/O to the hardware under test. The Simulink model is then loaded on to the xPC machine and is run in real time. The input signal is captured from the front microphone of the BTE microphone shells connected to the analogue board of the xPC machine.

xpc model: //P8825_SCORES/10_Sub_Projects/17_OVC/release/version 1/nsb_xPC_Own_Speech_Classifier_Model_V1.slx

Simulink model : //P8825_SCORES/10_Sub_Projects/17_OVC/release/version 1/Own_Speech_Classifier_Model_V1.slx

2.2.3 Classification

Once the features have been extracted, they are then sampled at different rates (2 Hz and 5 Hz define as the update rate), and then used to determine the most likely class. This class determination is done in classification block which uses the decision tree approach. The decision tree at this stage is chosen just because it is already being used in the Environmental Classifier and is computationally less expensive.

2.2.4 Decision Tree

A Decision Tree is a simple but effective classification method. The final system is intuitively structured like a tree, where a decision of which branch to follow is made at each node, based on whether a specified condition is met or not. In the type of Decision Tree we are using, the condition is always the comparison of the current value of a feature to a particular threshold. The leaf nodes in the tree are then the final classification. This is illustrated in Figure 24 below.

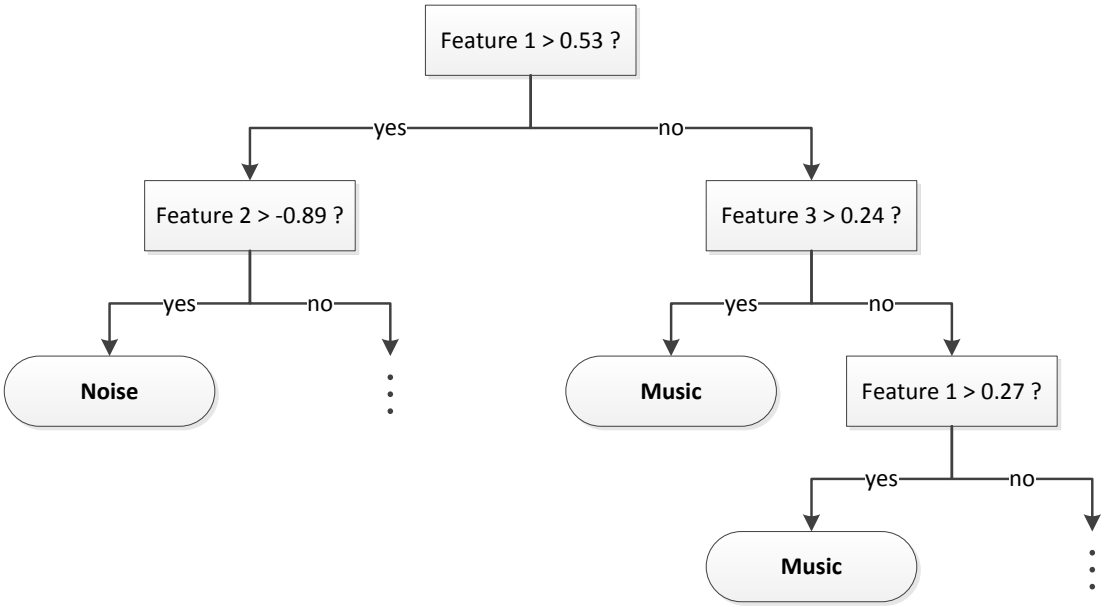

Figure 8: An example Decision Tree structure. At each node one of the features is compared to a given threshold. If it is greater than the threshold, traverse left, otherwise traverse right. Once a leaf node is reached, the classification is determined by the class at that leaf node.

Although traversing a tree to find the class for a particular set of input features is not difficult once the Decision Tree exists, determining what structure the Decision Tree should take on is more difficult. This process is called training. The idea behind training is to produce a tree that best classifies a set of known samples (the training database).

The decision tree was trained using Weka version 3.4 via FeatureGUI using Weka's J48 implementation of the C4.5 algorithm. Weka is publicly available software for Machine Learning and Data Mining produced by the University of Waikato, New Zealand[7]. This version is incorporated into FeatureGUI. The C4.5 algorithm works as follows (from Wikipedia [7]):

"At each node of the tree, C4.5 chooses one attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. Its criterion is the normalized information gain that results from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recurs on the smaller sub-lists."

"C4.5 goes back through the tree once it's been created and attempts to remove branches that do not help by replacing them with leaf nodes."

[7] Weka 3: Data Mining Software in Java. The University of Waikato, New Zealand. www.cs.waikato.ac.nz/ml/weka/. Last accessed on 30/10/2015.

For the prototype own voice classification Classifier, all features are extracted for the entire recorded sound database (see section **Error! Reference source not found.**) using the Simulink model and then processed via Weka (using FeatureGUI – see section **Error! Reference source not found.**) with 'Random Selection Training Data' and 'Percentage Training Data' set to 99 (i.e. use all of the training data). The resulting tree is then loaded to the Simulink model for the classification purposes. The results and the .arff files can be found in perforce.

2.2.5 Training

The final decision tree was trained on the recorded database which was also equalized for the level difference and then additive noise was also added to simulate the noisy conditions. It was found that on average the 'Own Speech' is 3-6 dB higher in level then the 'External Speech'. To avoid this heavy dependence of the classification on the level the decision tree is also trained on the dataset when there is no level difference between the 'own' and 'external' speech. For this purpose the data records were equalized for 'Own Speech' and 'External Speech' to have similar number of records for different levels. The probability density functions after the equalization are shown in figure 7.

Furthermore, to simulate the noisy conditions noise was added to the recorded database. The decision tree is trained on clean, 15 dB, 10 dB and 5 dB SNR. The dataset contains 60% of clean speech and 40% of noisy speech records.

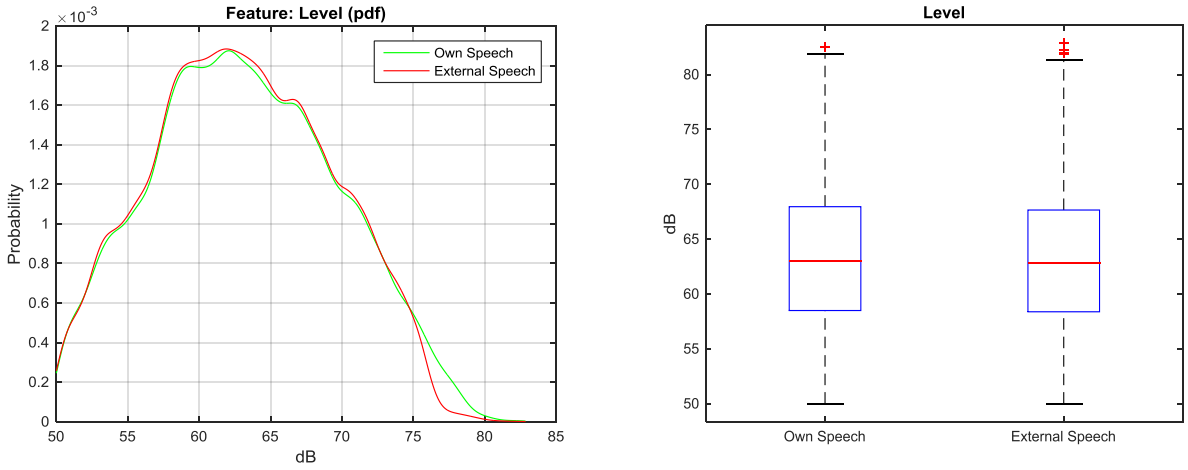

Figure 9: Probability density functions when the levels for the 'Own Speech' and 'External Speech' are equalized for recorded database without adding noise.

2.2.6 Results

The classification results based on all the 6 features are shown in figure 8 for an update rate of 2 on the training data set. The results are very similar when 66% of data is used as training and 33% is used for testing. Therefore for the final design all the data is used for obtaining the decision tree. The accuracy on the recording clean data set is close to 90% for both the 'Own Speech' and 'External Speech' classes. It is clear that accuracy goes down for own speech in noisy conditions.

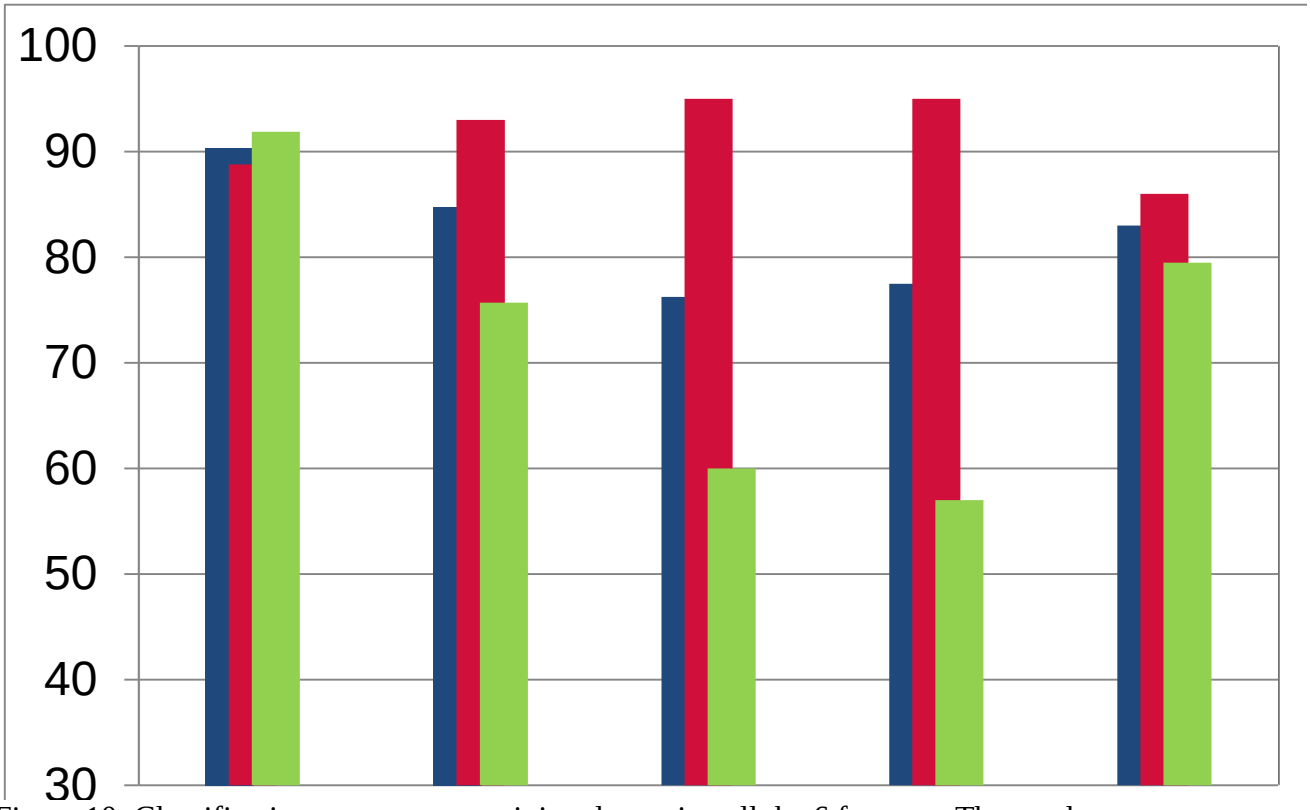

Figure 10: Classification accuracy on training data using all the 6 features. The results were very similar when 33% of the data was used as test set.

Confusion Matrix:

Clean Speech:

=== Confusion Matrix ===

a	b	c	<-- classified as	
8204	0	1039	a = 3	(‘External Speech’)
0	3383	0	b = 5	(Quiet)
787	0	8939	c = 6	(‘Own Speech’)

Over all training database (Clean + addition noise at three different SNR’s (SSN and MT))

=== Confusion Matrix ===

a	b	c	<-- classified as
134830	0	22034	a = 3
0	32108	0	b = 5
32751	0	126741	c = 6

2.2.7 Performance on a test conversation

The data logging performance on a real conversation (Benson_Obaid.wav) between two persons is shown below. It is noteworthy that the 'Own Speech' and 'External Speech' durations are close to the true values labelled manually by a human transcriber. Furthermore, as long as the time duration and turns taken are both close to true values then the underlying segmentations are also very similar.

Update rate →	100 ms	200 ms	500 ms	True Value
Own Speech	77.7	77.8	88.4	77.8
Quiet	22.3	24.4	11.2	23.8
External Speech	73.1	70.8	72.4	71.4
Turns taken	9	7	7	7/8

Figure 11: Data logging results for the test recording in speech environment (no noise). Numbers are time durations in seconds.

2.3 Phase 3: Real-Time platform implementation on Beaglebone

Up till the phase 2 the 'Own Voice Activity Detection' classifier works on the audio recordings or on the xPC system in the lab environment. To conduct the take-home trials we need to develop a small enough system which could be used in the home environments with a reasonable amount of effort. For this purpose the beagle board solution is chosen. Beagle boards are tiny computers with all the capability of today's desktop machines, without the bulk, expense, or noise. A working system is shown in the figure 12. Currently the implementation of the algorithm on the board is in progress. The accuracy of the classifier will be evaluated and necessary recommendations and optimizations will be implemented to fine tune the algorithm.

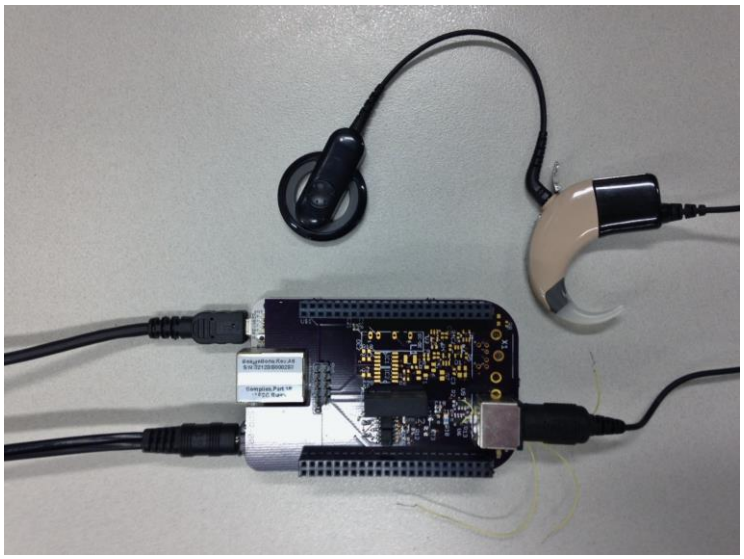

Figure 12: Beagle board runs the own voice classifier which is used for the take home trials.

3. CONCLUSIONS

An ‘own’ speech classifier is designed and evaluated to decide about the feasibility of an ‘own’ speech class for linguistic data logging. It is shown that ‘own’ speech can be classified with reasonably high accuracy from the ‘external/partner’ speech based on decision tree classifier on a reasonable database of recordings. Furthermore, 6 features are selected which can separate most of the instances in the training and test datasets. The main research questions that have been answered are as follows

- It is feasible to detect own speech from the external/partner speech with high accuracy (>80 %) with a classification tree approach in relatively low noise conditions.
- The computational complexity of the algorithm is reduced in phase 2 of the project to make it feasible to implement it on a real-time system.
- It is best to include three new features to the current environmental classifier design. Thus in total 6 features are enough to achieve higher level of accuracy in relatively low noise conditions.
- It is also recommended that the update rate of the classification tree should be increased enough to detect short words contribute (e.g yes, no kind of responses).
- The algorithm is implemented on the Beagle board real-time system for the take home use and evaluation. Currently the verification of the algorithm on the board is in progress.