Access Control Method for the Offline Home Automation System

Mikołaj Pabiszczak
Gido Labs
Poznań, Poland
Email: m.pabiszczak@gidolabs.eu

Monika Grajzer
Gido Labs
Poznań, Poland
Email: m.grajzer@gidolabs.eu

Łukasz Sawicki
LARS Andrzej Szymański
Niepruszewo, Poland
Email: lukasz.sawicki@lars.pl

***Abstract*—The paper proposes a Decision Support System for controlling access to the core functionalities of speech-based, embedded home automation devices working fully offline. The system is based on the Keyword Spotting technology recognizing custom keywords from a non-English language. Even though the performance of this technology in the scientific set-ups is relatively high, the reported false positive rates are still not sufficient for commercial application. Therefore, in our solution, we incorporate a number of computationally lightweight modules that considerably reduce the rate of false alarms while retaining an acceptable access rate.**

Keywords–voice-controlled home automation; Access Control; Decision Support System; Keyword Spotting.

I. INTRODUCTION

Home automation systems supplied with speech-based interfaces are becoming increasingly popular and are currently used to command a variety of home appliances, such as, e.g., thermostats or lighting [1]. In such systems, Automatic Speech Recognition (ASR), being a part of the core Dialogue System, is a resource-consuming task, which may use a significant amount of device resources – especially if implemented on embedded devices. Therefore, voice-controlled systems and applications are usually not continuously performing ASR, but rely on the Access Control method that activates this technology only when needed. Such a methodology constitutes a Decision Support System (DSS), which contains intelligence for analysing acoustic-based knowledge [2] – coming primarily from Voice Activity Detection (VAD) and Keyword Spotting (KWS) technologies. VAD detects when speech occurs in a signal collected by the microphone and switches the device to the listening mode, while KWS aims at detecting, if the particular keyword phrase has been spoken (such as, e.g., "Hey, Siri" or "Alexa"). Other methods of Access Control for home automation devices may include also Speaker Recognition (SR) technology, which is based on voice biometrics and allows only authorized users to gain access to the system [2].

Due to the complexity of speech processing, many components of the Access Control and ASR procedures are often performed in the cloud. Cloud-based audio processing may, however, raise privacy issues – the audio recording is stored at external premises and may potentially be reviewed by the third parties [3]. This creates a need for solutions working locally, fully offline. Developing such systems for embedded devices is challenging due to the scarcity of computational power and other resources, especially battery power. In addition, an Access Control solution should be characterised by a very low False Positives Rate (FPR) in order not to raise false alarms and not to grant unauthorised access to the core home automation system. Especially in the case of the KWS module, even though the most recent research results are very promising, the reported FPR levels are still not sufficient for commercial application when applied without any supporting intelligence [4]–[6]. Moreover, the KWS systems are typically aimed at recognising English speech/speakers [4]–[6]. In order to produce a system working for a less common language (such as, e.g., Polish), it is necessary to gather a training database of examples, which is both time-consuming and financially expensive. Providing a well-performing system with only a small-size database available is very challenging.

To address the above challenges, we propose in this paper a DSS that would work on an embedded device and allow to efficiently make decisions on granting access to the voice-controlled home automation system. The decision-making engine of this solution exploits the knowledge extracted from the audio signals – collected by the device microphone with the support of KWS and SR modules. A core part of the proposed Access Control DSS system is the KWS module and the related inference engine. Hence, in this paper, we primarily present our research on these technologies, focusing on the design of 1) a KWS module for recognising a keyword from a non-English language (in our case Polish), and 2) computationally lightweight procedures that address the problem of high FPR and allow to significantly reduce the number of false alarms in the test environment. The presented results depict the comparison of false positive and True Positive Rates (TPR) of different DSS schemes investigated by us while addressing the research problems identified above. In this paper we will focus on the Access Control design, nevertheless it should be noted, that this is an initial stage of the entire voice-based interface, which is typically followed by an ASR-based dialog system responsible for recognizing speech and for command understanding. The design of the these latter modules is, however, out of the scope of this paper.

The rest of the paper is organised as follows: in Section II, we present a short overview of related work, followed by the description of the proposed DSS system and the methodology taken to implement its components for embedded devices in Section III. In Section IV, we discuss in detail our results and, finally, in Section V we present conclusion and suggestions for further improvements as well as future work.

II. RELATED WORK

In the voice-controlled home automation systems, the devices offer user interfaces, which are designed to control an access to the core system functionality [7]–[9]. As the most straightforward solution, they are composed of VAD/KWS

 ISBN: 978-1-61208-765-8

module [5]. The KWS system, which has been trained on examples including silence or background noise samples, can also play the role of a VAD solution. This makes the KWS module a core part of the Access Control technologies. Most recent research on KWS systems focuses on applying neural networks – with particular interest in the Convolutional Neural Networks (CNNs) [4]. In this context, an increasing attention was given to the so called Residual Neural Networks architectures (ResNet), which incorporate skip connections between blocks of selected layers. This kind of architecture was first used to train extremely deep networks aimed at image recognition and was then exploited for the audio processing tasks, such as KWS and SR technologies [5][10]. ResNets are characterised by a lower complexity and faster training phase and were proved to obtain very good performance even for relatively small-sized networks – reaching the accuracy of 95% [5]. While such results are impressive, they are far from being industrially applicable: assuming that FPR is 2% [5] and the system makes a prediction every second, there will be 72 false alarms in one hour — a number unacceptable from the point of view of an end-user.

In addition, the KWS-based Access Control technology may also be accompanied by the SR system to form a biometric-based DSS, which can identify speakers and grant them proper permissions [11]. Nonetheless, even combining the modules together does not improve the FPR of the Access Control DSS as a whole. Some approaches to address this problem introduce pushing a button [8], detecting the audio louder then a certain threshold [7] or rely on more advanced features – such, e.g. in [9], where KWS is followed by additional reasoning using Hidden Markov Models and re-checking in the cloud. The latter kind of solutions, used, i.a., by Apple, Google or Amazon, are often very complex and likely too resource-consuming for small embedded devices. On the other hand, many related works on voice interfaces do not consider the methods of Access Control explicitly [12][13], neglecting the aspect that has tremendous impact on the practical implementation of such solutions. This calls for new approaches that would offer required effectiveness on embedded devices.

III. System Architecture

We are considering a DSS system that is controlling access to embedded home automation devices. It is working locally on the device without any access to the Internet and cloud-based resources. The proposed system architecture is depicted in Figure 1.

In the envisioned design, if the user wants to activate the home automation device, he/she has to say the specified keyword and wait for a signal (buzz), which will confirm granting an access to the core ASR-based dialogue system of this device. The signal will be generated only if the user 1) has spoken the correct keyword and 2) was authorized as one of the known users.

The straightforward approach to the design of an Access Control DSS for the embedded devices would incorporate only the blocks of KWS and SR marked dark-grey in Figure 1. Combined together, they grant system access to the authorized users who have uttered particular keyword phrase. In the course of our research, we have investigated introduction of additional subsystems (light-gray boxes in Figure 1) that would allow to decrease the FPR of the entire Access Control DSS, without a significant loss in its TPR.

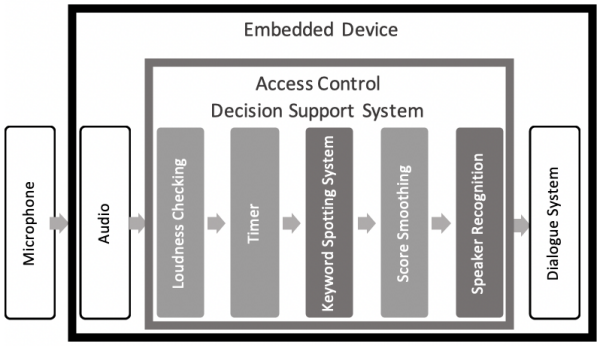

Figure 1. System design. Light-gray boxes represent new modules added in the course of research.

The first new element – "Loudness Checker" – requires the incoming audio to have a certain minimum level of loudness. The second one, called "Timer", limits the time duration of an audio buffer that is being processed – if the triggering word is not observed shortly after that, it may be assumed that the audio resulted from some background noise or other conversational sounds and does not need to be processed further. Such design also allows to lower the resource consumption, since the predictions performed by the KWS and SR modules are no longer executed continuously. The third system – "Score Smoothing" – performs the smoothing of the results obtained from KWS.

These decision-making subsystems together form an Access Control DSS, where at each step decisions are made based on the knowledge collected on the previous steps. The detailed description of each element is presented in the following subsections.

A. Keyword Spotting module

The KWS system is based on the neural network with ResNet architecture "res8" from [5], which is characterised by a small number of network parameters (approx. 110K). This architecture is comprised of 3 residual blocks, each containing: convolutional layer with ReLU activation, Batch Normalization, another convolutional layer similar to the first one and the second Batch Normalization. Between these blocks skip connections are present (the so called residual connections). Such a sequence of blocks is preceded by a convolutional layer with ReLU activation together with an average pooling layer. The output of last residual block is fed into an average pooling layer, flattened and fed into dense layer of size corresponding to the number of labels that can be recognised.

The system in [5] was trained for the English keywords. Since our system is targeting Polish language speakers and the keyword itself is atypical (a trade name of the product of a polish company), the challenge was to produce a well-performing classifier having available only limited database of related audio samples. Hence, a procedure called "transfer learning" was used. In this approach an already-trained neural network targeting a similar problem was used as a "feature extractor" and only its last layer (i.e. the classifier) was trained by using a smaller, problem-specific dataset. In our case, a "res8" network [5] was used as a feature extractor, which was trained on the Google Speech Commands Dataset (GSCD)

 ISBN: 978-1-61208-765-8

for recognizing 10 keywords from this dataset and the labels of "silence" and "unknown". In order to train the classifier layer of our solution, with 2 outputs (keyword vs. non-keyword), we have gathered a dataset of 698 positive examples of the selected keyword from 36 people. For the negative examples, we have recorded a larger database of phonetically representative Polish speech. In addition, for the training phase 80% of examples were augmented with background noises of variable loudness, originating from the GSCD dataset. These background noises were also used to create negative examples of silence (i.e., the lack of speech).

The input of the neural network requires $1s$ of audio. Hence, for the system working in real-time, a sliding window of $1s$ that moves by $0.2s$ is used (i.e., the audio for the two adjacent predictions coincides in 80%). Those parameters were established experimentally, as a compromise between satisfactory spotting of utterances and computational burden. 40 Mel-Frequency Cepstral Coefficients acoustic features are then calculated for such a window, normalised, and fed into the ResNet, which acts as a binary classifier returning a numerical score. The score may be considered as the probability that the analysed audio snippet contains the desired keyword. Typically, if the score is higher than a certain threshold, it is assumed that the keyword was uttered. In our case, a threshold corresponding to the Equal Error Rate (EER) was taken, which was calculated with the use of a Receiver Operating Characteristic (ROC) curve and its Area Under Curve (AUC) obtained for the data in the test set. This way the operating point of our system (neural network) was specified.

B. Additional Modules

The Loudness Checking module is the first block of the proposed Access Control DSS, which constantly reads the audio input from the microphone and checks its mean amplitude. If it is above a certain threshold, the audio from the microphone is fed into neural network classifier. This threshold was set experimentally, as it depends on: 1) the particular microphone used, 2) the format of the audio encoding, and 3) a microphone driver.

The Timer module, directly following the Loudness Checking, processes only the signals which are considered strong enough. Since the duration of the desired keyword is typically not longer then $1s$, the timer module limits the audio input, being processed by the KWS model, to the first $1.2s$ of the signal (the lacking part of the $1s$ sliding window in the beginning of listening is filled with zeros). With the overlapping between consecutive frames of 80%, this gives the input to the KWS module of maximum 7 frames, which constitutes an attention window of size 7.

The third module, named Score Smoothing, contains the final evaluation engine and is introduced directly after the KWS block. It implements an additional logic introduced to extract the most meaningful knowledge from the KWS module output. Based on our observations (see Section IV), the FPR is too high if an access is granted after observing only a single frame containing the keyword. Therefore, the Score Smoothing module aims at smoothing the scores of the KWS classifier. Based on the observations from the initial field trials, our approach for this block is based on calculating the mean of the last n predictions – in the beginning of listening, when less number of predictions is available, the lacking results are assumed to have score 0 (we experimented with n equal to 3 and 4). This way, a single, strong trigger could still be considered as a positive activation. The Score Smoothing module makes a decision based on the observed mean value – once it exceeds the threshold set for KWS, the input audio signal is put for further processing by the SR module.

C. Speaker Recognition module

The SR module takes as input a single frame, which is fed from the Score Smoothing module as the one with the strongest trigger. The system is based on the ResNet neural network – following the architecture specified in [10] and the related model, which was further trained by us with low learning rate, on the database containing voices of 100 polish speakers. In the test on the database of 36 polish speakers (Section III-A), where a single speaker was identified, the EER of the new model was 1.7% (comparing to 2.27% of the original one by [10]). If one of the known speakers is found, the system grants the access to the ASR-based dialogue system.

IV. Evaluation

We performed the evaluation of the 3 main components, which were introduced to the proposed Access Control DSS (light grey in Figure 1) – with the aim to estimate their influence on both FPR and TPR measures. We report those measures, since they present the system performance in the operating point, which was specified by the selected system set-up (in case of the KWS module – by the threshold corresponding to the EER). The focus of the executed trials was on the new modules, which closely cooperate with the KWS module – further integration with the SR module is a part of the future work. We performed two tests in diversified conditions: the first one was used to determine the influence of the various components and their combination on the false positive activations caused by the background voices, which may occur in the household. The aim of the second test was to identify the influence of the selected designs on the TPR and false alarms in a challenging task, where speech samples included words that are phonetically similar to the keyword or are household-related.

The entire DSS system was implemented on a RaspberryPi 3B (CPU: 1,2 GHz quad-core ARM-8 Cortex-A53 (64-bit); 1 GB RAM). The device was equipped with a custom-made microphone matrix with 5 independent microphones. The experiments were conducted as field trials where real hardware was used. Unless stated explicitly, the parameters of the other modules were set as described in Section III.

A. FPR measures for different DSS designs

The aim of the first test was to estimate the FPR of the enhanced Access Control DSS, consisting of various combinations of proposed subsystems, in comparison to the standard Access Control method based only on the single KWS module. During the trials, 18min and $14s$-long audio of varying loudness was analysed. Radio conversations were chosen as the audio source, containing the voices of various people. The activation keyword was not present in the audio recordings. The DSS system was processing this audio and the number of false alarms was counted at the output – reflecting the number of positive access activations made after the system has wrongfully detected the keyword. FPR was calculated as the ratio of the number of these false alarms to the number of all analysed audio frames (equal to 5471).

 ISBN: 978-1-61208-765-8

TABLE I. FPR VALUES FOR VARIOUS ACCESS CONTROL DSS DESIGNS.

Design	*FPR [%]*
Reference system - only KWS	**2.23**
Loudness Checking	0.90
Loudness Checking + Timer	0.64
Score smoothing – avg. over 3 last frames	1.43
Score smoothing – avg. over 4 last frames	1.30
Loudness Checking + Timer + Score Smoothing [avg.3]	**0**
Loudness Checking + Timer + Score Smoothing [avg.4]	0

The results of this test can be found in Table I. For a reference system containing only the KWS module, FPR of 2.23% is obtained. Adding a Loudness Checking module allows to reduce FPR 2.5 times – to the value of 0.9%. This result is, however, still not satisfactory for commercial applications. The additional use of a Timer module enables to reduce the number of false alarms by a factor of 3.5 (in comparison to the reference system) and to obtain FPR of 0.64%. For the combination of KWS with a Score Smoothing module, using the average of the last 4 predictions gives the FPR of 1.3% and the average of the last 3 – 1.43%. In case of the system for which all three modules were incorporated into one processing pipeline (in two variants – with Score Smoothing using 3 and 4 predictions) the number of false alarms is reduced to zero. The results show that this set-up is the most promising one.

B. Estimation of DSS system performance

The aim of the second test was to obtain the values of TPR, FPR and accuracy for the key Access Control DSS designs. Among the 3 new modules, the Score Smoother is the one that can potentially have negative impact on the TPR measures. Therefore, we have been investigating which configuration – with averaging over 3 or 4 frames – will result in better overall performance. In this trial the audio signals were recorded live from 13 users (both female and male). For each person, the test consisted of uttering the keyword 30 times and uttering 10 other words 3 times each. Approximately 20% of them were phonetically similar to the keyword, which makes this trial particularly challenging. For each utterance, the binary result assigned by the Access Control DSS system at the Score Smoother output was recorded.

TABLE II. TPR, FPR AND ACCURACY FOR DIFFERENT ACCESS CONTROL DSS DESIGNS.

Design	*TPR [%]*	*FPR [%]*	*Acc. [%]*
Reference system - only KWS	90.77	5.90	92.44
Score smoothing [avg.3]	**86.41**	**4.87**	**90.77**
Score smoothing [avg.4]	84.10	4.87	89.62

The results are presented in Table II. The reference set-up with only KWS module present obtains TPR of 90.77% and FPR of 5.9% with overall accuracy of 92.44%. Such a result can be considered comparable with the results reported in the literature (e.g., approx. 95% in [5]) – this is a very good result bearing in mind that the neural network was trained on a smaller set of examples and that it was tested on the dateset containing also the words phonetically similar to the keyword. The DSS design with the Score Smoothing module using the last 3 predictions gives overall better results than the one with 4 predictions – obtaining TPR of 86.41% and FPR of 4.87% with accuracy of 90.77%. Hence, we have decided to include this variant in the final system design.

This final system set-up was also assessed towards the imposed delay – with regard to the KWS-related functionality (the detailed analysis of the SR module is part of the future work). Within the proposed Access Control DSS, the main component introducing delay is the Keyword Spotting System (please refer to Figure 1), since it consumes much more computational resources than the other DSS blocks. Hence, their delay can be assumed negligibly small. For the KWS module, the obtained delay is measured as the time between the moment of loading the audio from a microphone buffer and the moment of obtaining the result of prediction. In the investigated set-up, it varies between 0.4-0.5s, which is small enough to be considered practically applicable.

C. Lessons learned

Even though the second trial has shown the decrease in the TPR for the final DSS system variant of approx. 4% in comparison to the reference set-up, we consider it to be still applicable and acceptable, taking into consideration the very positive impact on the reduction of false alarms observed for this solution in the first trial. In addition, we have noticed in the course of conducted experiments that the observed False Negatives corresponded mostly to a lousy articulation/quiet speech, which could have been improved by the speakers and yield better results in the repeated trials. For an FPR reported in the second trial, which is still high, it is important to remember that a fifth of the testing words was phonetically very similar to the keyword and, as a result, the performed test was a challenging one. In general, the negative examples in this trial rarely occur in the beginning of the speech, and the chance of finding them in the attention window set out by Timer module is low. Thus, the FPR reported in the trial may be thought of as a "worst-case scenario". Considering the obtained results, we have decided to choose for the final DSS system the design for which the Score Smoothing module is using last 3 predictions and the system is equipped with Timer and Loudness Checking modules.

V. CONCLUSION AND FUTURE WORK

We have proposed a DSS system, which grants access to the voice-controlled home automation devices. It was introduced in order to decrease the FPR and diminish the number of wrongful activations while the home automation device is continuously processing the collected audio signals. For this purpose, 3 modules of the new DSS system were proposed to accompany KWS and SR technologies, which are typically used to activate an ASR-based dialog system only if the specified keyword is pronounced by the known system users. The performed evaluation enabled to assess these candidate solutions and select the best performing variant. The selected design allowed to considerably reduce the FPR of the entire DSS system while retaining acceptable TPR and keeping overall accuracy above 90%. Substantially, in the performed trails, the proposed solution allowed to entirely suppress false alarms caused by background radio voices, while the reference set-up generated approx. 122 unwanted activations per 5471 analysed frames. As a result, with the proposed computationally lightweight modifications, we have come up with an Access Control DSS that is commercially applicable.

 ISBN: 978-1-61208-765-8

As a part of the future work, we plan on further improving the system efficiency by experimenting with substituting the Loudness Checking subsystem with more sophisticated VAD. With a larger amount of available user data, we are also considering the possibility to substitute the output stage of Score Smoother with a trained classifier. Moreover, we envision to perform an evaluation of the entire Access Control DSS, including the SR module, in realistic set-ups with possibly high number of end-users.

ACKNOWLEDGEMENT

The research was supported by the National Centre for Research and Development in Poland under the grant no. POIR.01.01.01-00-0044/17

REFERENCES

[1] C. J. Baby, F. Khan, and J. N. Swathi, "Home automation using web application and speech recognition," in 2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS), 2017, pp. 1–6.

[2] A. Kaklauskas, Intelligent Decision Support Systems. Springer International Publishing, Cham, 2015, ch. 2, pp. 31–85, in Biometric and Intelligent Decision Making Support, ISBN: 978-3-319-13659-2.

[3] A. Cuthbertson. Google defends listening to private conversations on google home: But what intimate moments are recorded? [Online]. Available: https://www.independent.co.uk/life-style/gadgets-and-tech/news/google-home-recordings-listen-privacy-amazon-alexa-hack-a9002096.html [retrieved: 11, 2020]

[4] T. N. Sainath and C. Parada, "Convolutional Neural Networks for Small-footprint Keyword Spotting," Proc. 16th Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH 2015, 2015, pp. 1478–1482.

[5] R. Tang and J. Lin, "Deep residual learning for small-footprint keyword spotting," in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018, pp. 5484–5488.

[6] C. Lengerich and A. Hannun, "An end-to-end architecture for keyword spotting and voice activity detection," arXiv preprint arXiv:1611.09405, 2016.

[7] C. Sidhartha, S. Siddharth, S. S. Narayanan, and J. H. Prasath, "Voice activated home automation system," in National Conference on Man Machine Interaction 2014, vol. 1. ASDF, India, 2014, pp. 69–72.

[8] K. A. Lee, A. Larcher, B. Thai, B. Ma, and H. Li, "Joint application of speech and speaker recognition for automation and security in smart home." in INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, 01 2011, pp. 3317–3318.

[9] Siri Team. Hey siri: An on-device dnn-powered voice trigger for apple's personal assistant. [Online]. Available: https://machinelearning.apple.com/research/hey-siri [retrieved: 11, 2020]

[10] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 5791–5795.

[11] M. Vacher et al., "Speech and speaker recognition for home automation: Preliminary results," in 2015 International Conference on Speech Technology and Human-Computer Dialogue (SpeD). IEEE, 2015, pp. 1–10.

[12] Y. Mittal, P. Toshniwal, S. Sharma, and D. Singhal, "A voice-controlled multi-functional smart home automation system," in 2015 Annual IEEE India Conference (INDICON), 10 2015, pp. 1–6.

[13] G. López, V. Peláez, R. González, and V. Lobato, "Voice control in smart homes using distant microphones: A voicexml-based approach," in Ambient Intelligence, D. V. Keyson et al., Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 172–181.

 ISBN: 978-1-61208-765-8