ORIGINAL RESEARCH
published: 13 May 2022
doi: 10.3389/fcomp.2022.778367

Speech Assistant System With Local Client and Server Devices to Guarantee Data Privacy

*Hans-Günter Hirsch**

Institute for Pattern Recognition, Niederrhein University of Applied Sciences, Krefeld, Germany

Users of speech assistant systems have reservations about the distributed approach of these systems. They have concerns that people might get access to the transmitted speech data or that somebody is able to access their microphone from outside. Therefore, we investigate the concept of a setup with local client and server systems. This comes along with the requirement of cost-efficient realizations of client and server. We examined a number of different cost-efficient server solutions depending on the required recognition capability of specific applications. A fairly cost-efficient solution is the use of a small computing device for recognizing a few dozens of words with a GMM-HMM based recognition. To perform a DNN-HMM based recognition, we looked at small computing devices with an integrated additional graphical processor unit (GPU). Furthermore, we investigated the use of low-cost PCs for implementing real-time versions of the Kaldi framework to allow the recognition of large vocabularies. We investigated the control of a smart home by speech as an exemplary application. For this, we designed compact client systems that can be integrated at certain places inside a room, e.g., in a standard outlet socket. Besides activating a client by a sensor that detects approaching people, the recognition of a spoken wake-up word is the usual way for activation. We developed a keyword recognition algorithm that can be implemented in the client despite its limited computing resources. The control of the whole dialogue has been integrated in our client, so that no further server is needed. In a separate study, we examined the approach of an extremely energy-efficient realization of the client system without the need of an external power supply. The approach is based on using a special microphone with an additional low-power operating mode detecting the exceeding of a preset sound level threshold only. This detection can be used to wake up the client's microcontroller and to make the microphone switch to normal operating mode. In the listening mode, the energy consumption of the microphone is so low that a client system can be active for months with an energy supply from standard batteries only.

Keywords: speech assistant, local recognition, compact client, keyword recognition, energy efficient client

OPEN ACCESS

Edited by:
Benjamin Weiss,
Technical University of Berlin,
Germany

Reviewed by:
Stefan Schaffer,
German Research Center for Artificial Intelligence (DFKI), Germany
Matthias Wolff,
Brandenburg University of Technology Cottbus-Senftenberg, Germany

Correspondence:
Hans-Günter Hirsch
hans-guenter.hirsch@hs-niederrhein.de

Specialty section:
This article was submitted to Human-Media Interaction, a section of the journal Frontiers in Computer Science

Received: *16 September 2021*
Accepted: *14 April 2022*
Published: *13 May 2022*

Citation:
Hirsch H-G (2022) Speech Assistant System With Local Client and Server Devices to Guarantee Data Privacy. Front. Comput. Sci. 4:778367. doi: 10.3389/fcomp.2022.778367

1. INTRODUCTION

Right now, the way to omnipresent speech assistants is determined by special hardware realizations like the Echo devices by Amazon or the Google home devices as well as by software realizations like Siri by Apple or Cortana by Microsoft. Most of these solutions consist of three components. The first component contains the hardware with microphones and loudspeaker to record and playback speech. This component is referred to as client. The recorded speech signal is preprocessed in the client to reduce the effects of background noise and reverberation. This is usually done by recording the speech with several microphones and applying multi-channel processing techniques. Furthermore, an algorithm is implemented in the client to perform the detection and recognition of a keyword that is used to wake up the assistant.

After wake-up, the preprocessed speech signal is usually transmitted *via* IP through a public network to a speech recognition server, which represents the second component of the entire system. The advantage of this approach is the application of a server configuration with extremely high computational performance, so that powerful recognition algorithms can be applied to enable high recognition performance. However, the data are transported to an external server *via* a public network, which means that it is not clear who gets access to the signal and what the signal could be used for besides its input to the recognition system. Furthermore, users have concerns that somebody can get access to their microphone from outside and can record and analyze audio when the speech assistant is not active (Chung et al., 2017; Lau et al., 2019; Malkin et al., 2019; Hernández Acosta and Reinhardt, 2020). The strength of this concern varies and depends on cultural and country-specific behavior of people. This leads to a high percentage of people in certain countries unwilling to use such systems, although they are not refusing speech technology in general. A number of approaches have been developed and investigated for the case of speech transmission through a public network. An overview about the privacy-by-design technology is given in Nautsch et al. (2019). The encryption of data (Nautsch et al., 2018; Bäckström et al., 2020) is an obvious approach to reduce the concern that somebody else besides the receiver can get access to the speech data or the recognition result. The binarization and protection of i-vectors (Mtbiaa et al., 2021) is an example for a so called cancelable biometric system. Speaker de-identification (Bahmaninezhad et al., 2018) represents another approach to privacy preservation. Furthermore, hardware based techniques can be applied like the software guard extension in Intels processor units (Brasser et al., 2018).

The result of the speech recognition is sent to a third component, e.g., as a text string. The dialogue between user and speech assistant is controlled by this additional server component. This server takes over several tasks. The first task is the interpretation of the received text string to find out what the user wants to know or intends to do. For example, if a user has uttered the sentence "What will be the weather in Krefeld tomorrow?" and if the recognition component has perfectly recognized the sequence of words, the task will be the correct interpretation of this word sequence to enable the search for the desired information. The inquiry can be formulated in many different ways, so that powerful natural language processing is needed at this point. The next task will be the acquisition of the requested information from a data base or another server. Then, a sentence has to be formulated as answer containing the acquired information. The sentence is further transferred to speech with means of a "Text-to-Speech" (TTS) algorithm. Finally, the speech signal is sent to the client to create an acoustic output as feedback to the user's inquiry. The example described before contains the retrieval of information as a frequent task given to a speech assistant. Besides this, users want to apply the speech assistant to control hardware devices at home or at a certain location. In this case, the third component may not need to or does not only have to create acoustic feedback. Its main task is the creation of a command that has to be sent to the hardware device. Often, this is not possible *via* direct communication between the dialogue component of the assistant system and the hardware device. Instead, another server system is needed that has access to and is able to control the hardware components.

In our investigation, we focus on the application of speech assistants to control hardware devices. This can be, for example, the control of a smart home environment. Our approach differs from the behavior of most commercial systems as they have been described before in two respects. First, we are investigating the concept of a setup with local client and server systems, which makes it possible to guarantee users that their speech is not transmitted outside their private networks and that nobody can access their microphones. To realize this concept, several requirements must be fulfilled, the main one being the application of a recognition server system that is affordable for private users on one hand and that guarantees a fairly high recognition performance on the other hand. As a second point, we want to simplify the whole structure of the system. Usually, commercial systems have to include the client and three server systems as described before. The possibility but also the difficulty are presented in Seiderer et al. (2020) to set up such a configuration as a local system with several open source components. Due to developing and integrating the needed components including the speech recognition ourselves we achieve a more compact and more flexible configuration in comparison to combining available open source components (Seiderer et al., 2020). Besides the recognition server, two additional servers are needed, one of them for the dialogue control including the speech interpretation and the other for accessing the hardware components. We combined the dialogue control and the communication with the hardware components in the client system. Thus, we can reduce the system to only two components including a client and a recognition server.

2. SYSTEM OVERVIEW

Figure 1 gives an overview of the goal of our project and the corresponding structure of the system. We want to allow the control of devices like shutters, lights, and any other type of

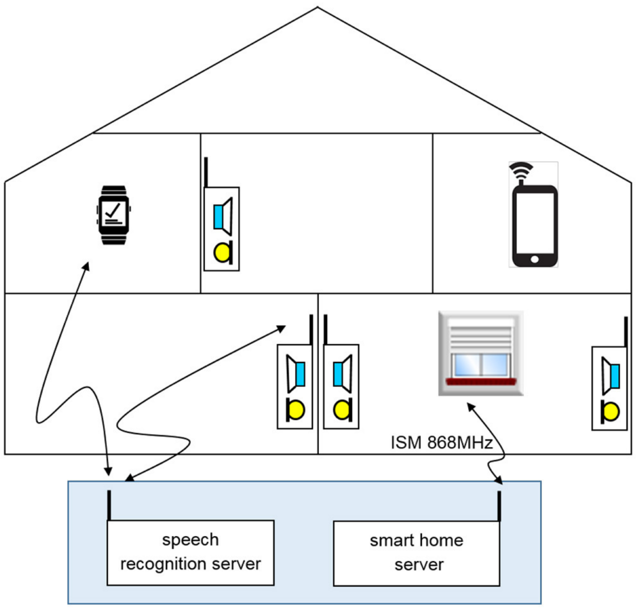

FIGURE 1 | Smart home control with local client and server systems.

actor by speech input from all rooms or locations in or nearby a building.

Therefore, we developed client systems with the capability of speech input and output that can be installed in each room. Our intention was to integrate these client devices in the existing electrical infrastructure as far as possible, e.g., as substitute of switches in existing outlet sockets. For the case that it is not possible to integrate them inside a room, we developed stand-alone client devices that do not need external energy supply. This leads to the requirement of an extremely low energy consumption, so that we investigated this aspect in a separate study. To cover also rooms or locations in the building that are rarely used or have almost no electrical infrastructure, we developed a software component for smart phones or smart watches where this component offers the same functionality as other client devices. Each client device can communicate with the in-house recognition server *via* LAN or WLAN. We look at the alternative of communicating and transferring speech *via* the DECT-ULE (Digital Enhanced Cordless Telephony- Ultra Low Energy) standard (ETSI, 2019) at a later point in this paper. Each client includes a dialogue control module, so that the device does not only accept the input of a single spoken command and plays back an acoustic reaction but can also manage a longer speech dialogue with the user. The client can send control commands to the local smart home server *via* LAN/WLAN. Furthermore, the direct switching *via* a relay is included in case of an integrated device as substitute for an existing switch. Cost and energy efficiency are the two main requirements for the design of the client systems. The target of low costs is also the main requirement for the choice of the local speech recognition server. We investigated different possibilities for realizing the speech

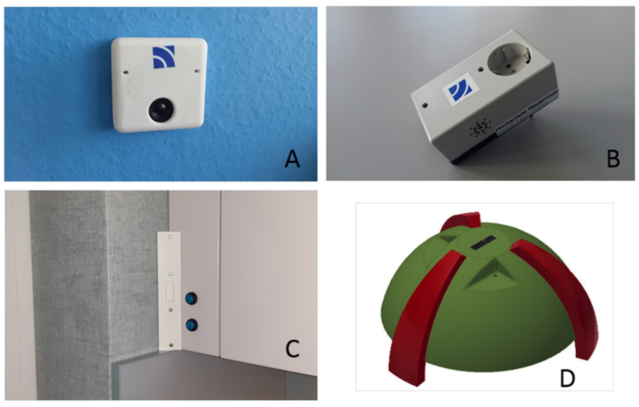

FIGURE 2 | Implementation of client devices in an outlet socket **(A)**, as a switchable power socket **(B)**, in a furniture corner **(C)**, and as a stand-alone device **(D)**.

recognition server depending on the demand of the application-specific recognition task. We applied our own realizations of phoneme-based GMM-HMM or DNN-HMM based recognition schemes when the recognition of smaller vocabularies containing up to a few hundred of words is needed. For cases in which the recognition task demands a larger vocabulary, we applied a Kaldi based recognition scheme (Povey et al., 2011) on a low-cost computer.

3. CLIENT

The goal of our investigation is the development of very compact client devices that fulfill the requirements of low cost and low energy consumption and that can be integrated in the existing electrical infrastructure of a building. After presenting the hardware setup, we will focus on the implemented software modules, especially the algorithm for detecting and recognizing the wake-up word. Furthermore, we will present the results of two studies to minimize the energy consumption of a stand-alone client and to communicate with a recognition server *via* DECT-ULE.

3.1. Hardware

Some of the devices we have developed are shown in **Figure 2**. **Figure 2A** shows the version of the client that can be integrated in an outlet socket. It can be taken as substitute for an existing switch. **Figure 3** is a block diagram containing all components of this version.

The small computer PI-Zero is used as a basis for this client and a separate small device with two microphones (Seeed, 2021) is applied to record the audio signal. A class-D amplifier is connected to the analog output of the sound device to allow audio output through the loudspeaker. Furthermore, a relay has been integrated that realizes the switching in case the client serves as substitute for an existing switch. The WLAN interface of the PI-Zero is used for communication with the recognition server. **Figure 2B** shows the integration of the client in a switchable

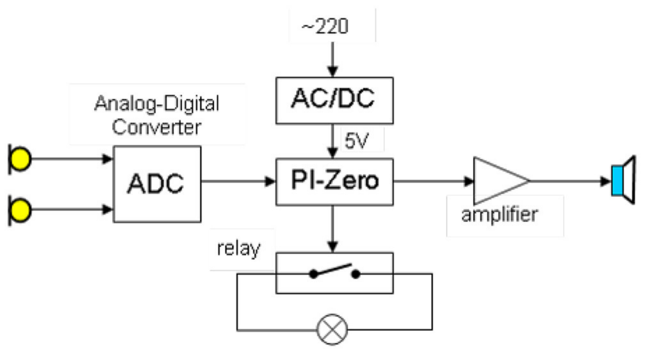

FIGURE 3 | Components of a client device.

power socket. The hardware setup is similar to the one shown in **Figure 3** but it contains two loudspeakers at the sides of the housing. **Figure 2C** shows the housing for an array of four microphones (Seeed, 2021) that is placed at the corner of kitchen furniture. A PI computer device is placed in the cavity behind the cover element in this furniture corner. **Figure 2D** shows a stand-alone version of the client, which is again based on a PI computer. The speech is recorded by an array of four microphones. The system is provided with energy by a rechargeable power bank. Later in this chapter, we will present results of a study where we investigated the application of a special microphone in combination with specially designed algorithms to setup a client system with extremely low energy consumption.

3.2. Dialogue Control

Most client devices run on Linux. The main software component is responsible for controlling the dialogue between user and client. The dialogue is modeled as a finite state machine. The states and their chronology are described by a text file, so that the dialogue can be easily defined and modified. An action is assigned to each state such as speech input, speech output, or sending a command sequence to a hardware device in the building or to the smart home server. This setup allows the individual configuration of each client, so that the dialogue can be defined depending on the room or the location of the client. For this application, we do not need a text-to-speech component in our client systems. Speech output is realized with pre-recorded audio files. The activation of the dialogue is an important feature of the client. We investigated the usage of sensors in our first versions to detect an approaching person. These sensors use ultrasound or infrared as basis for the detection. The integration and application of such sensors is useful in rooms or at locations where it is very likely that a person approaches the place with the intention to control a hardware device. In general, the recognition of a spoken keyword is the typical way of activating the client. Often, the keyword is called the "wake-up" word. The recognition of the wake-up word could be realized through the server system. However, this would usually lead to a high data traffic from the client to the server due to the permanent speech transmission or at least the transmission of the speech signal in all segments where a Voice Activity Detector (VAD) on the client assumes the presence of speech. Alternatively, certain stages of the recognition process could be distributed between client and server. In our investigation, we realized the recognition of the wake-up word in the client system, which requires an algorithm that can be implemented despite the limited computational performance.

3.3. Keyword Recognition

The performance of keyword recognition can be measured by the false acceptance rate (FAR) and the false rejection rate (FRR). In our application, where we want to apply the keyword detection for the activation of a home automation system, we prioritized lowering the FAR. Hence, we have to avoid any command recognition after an erroneous keyword detection because this could lead to the uncontrolled activation of devices at home. We developed an algorithm for the detection and the recognition of the keyword that consists of two stages as shown in **Figure 4**.

To extract characteristic features, the speech signal is sampled at a rate of 16 kHz. The short-term DFT spectra are calculated for frames containing 400 samples (=25 ms) every 10 ms after applying a pre-emphasis filtering and weighting the 400 samples of each segment with a Hamming window. The 400 filtered and weighted samples are transformed by means of a DFT with a length of 512. The short-term logarithmic energy logE is calculated by taking the logarithm of the sum of the squared DFT magnitude coefficients in the range between 200 Hz and 7 kHz. Furthermore, 12 cepstral coefficients C1–C12 are determined by transforming the logarithmic MEL spectrum with a DCT. Thirty-six MEL filters have been defined in the range from 200 Hz to about 7 kHz to calculate the MEL spectrum from the magnitude DFT coefficients. The Delta coefficients (ΔlogE, ΔC1, ..., ΔC12) and the second derivative of the energy contour $\Delta\Delta$logE are calculated according to the filtering scheme defined in ETSI (2003). The vector containing the 26 components (C1, ..., C12, ΔlogE, ΔC1, ..., ΔC12, $\Delta\Delta$logE) is used as feature vector for the GMM-HMM recognizer. The energy coefficient logE is omitted due to its varying value in case of background noise.

As the keyword, we chose the personal Name "Esmeralda," which is rarely used as a given name in Germany and which rarely occurs in German conversations. The word begins and ends with a vowel. It contains a fairly long sequence of phonemes, which supports a better recognizability. We created two Hidden Markov Models that represent the keyword and which are applied for the realization of the first recognition stage. The first model was created from real recordings of the keyword and from augmented versions of these recordings with HTK (Young et al., 2006). For data augmentation, we applied a tool (Hirsch and Finster, 2005) to create versions containing noise and reverberation, as these occur in real scenarios when recording in hands-free mode. The second keyword HMM is built as a concatenation of the corresponding triphone HMMs from a phoneme-based recognizer that has been trained on several hundred hours of German speech with HTK. The sequence of feature vectors and the keyword HMMs are taken to set up a GMM-HMM recognizer as first stage of the recognition algorithm. A set of 25 monophone HMMs and some pause and noise models are included as so-called filler models (Rose and Paul, 1990). The intention is the

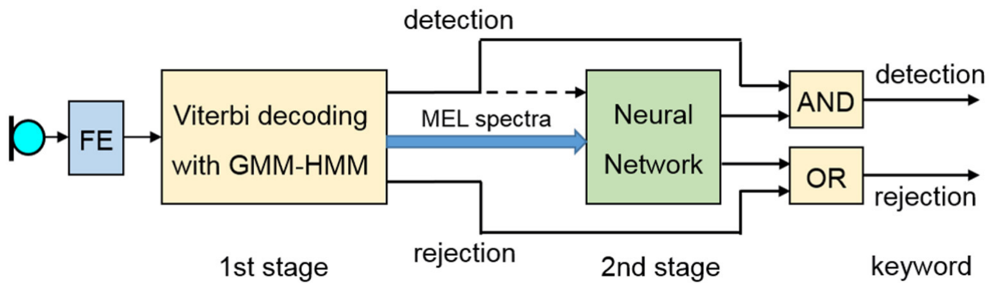

FIGURE 4 | Two-stage keyword recognition (FE, feature extraction).

modeling of speech not containing the keyword as sequence of filler and/or pause HMMs with a higher probability as with one of the keyword HMMs. Only the spoken keyword should lead to a higher probability when modeling it with one of the keyword HMMs. We implemented a fast notification at the end of a spoken keyword, so that further processing is not delayed. In case the GMM-HMM recognition stage indicates the detection of a keyword, we try to verify this assumption by a second stage. It is known that the modeling of speech by means of filler models works quite well, but the expected FAR of this approach is too high for the intended application. Therefore, we apply a neural network as the main component of the second stage. As input, we use the sequence of logarithmic MEL spectra within the speech segment that should contain the keyword according to the recognition result of the first stage. To get a fixed number of input coefficients for the neural network, we reduce the number of Mel spectra to 50 by iteratively calculating the mean of two consecutive logarithmic MEL spectra with the lowest spectral City block distance. **Figure 5** presents the average spectrum that has been calculated over 850 utterances of the keyword used to create one of the keyword HMMs. We observe a very characteristic spectral pattern where the spectral characteristics of each individual phoneme become clearly visible.

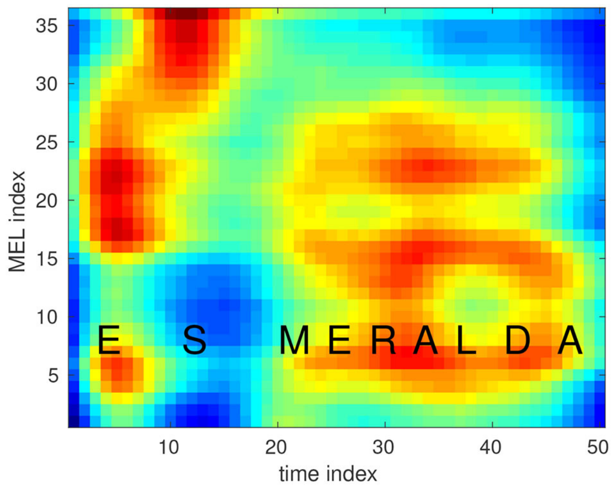

FIGURE 5 | Average MEL spectrogram of the keyword.

The input for the neural network consists of 1,800 spectral amplitudes from 50 spectra with 36 MEL coefficients. We apply a mean and variance normalization to each spectral pattern by calculating the mean and the variance over all 1,800 spectral parameters of each individual pattern. We apply a fully connected multi-layer perceptron, consisting of three layers. The first layer consists of 200 nodes, the second of 50 nodes and the output of two nodes for the two cases of a keyword and a non-keyword. To train the weights of the neural network, we needed spectral patterns for spoken keywords as well as for segments where the keyword was erroneously detected by the first stage. About 850 spectral patterns of the spoken keyword could be determined from the utterances that have been used for training the keyword HMM. To get spectrograms of speech segments where the keyword was not spoken, we applied the detection algorithm of the first stage to German speech data from different databases (Burger and Schiel, 1998; Radeck-Arneth et al., 2015). Several thousands of segments were erroneously detected. Thus, we had about 850 examples of the keyword spectrogram and several thousand examples of the non-keyword spectrogram available. We applied the tools of Chollet (2015) to estimate the weights of the neural network. We achieved a FAR of less than one keyword per hour of speech as the result of these simulation experiments (Hirsch et al., 2020). Then, we implemented and ran the algorithm for the keyword recognition on some of the client devices for several weeks in a laboratory and in a living room. False detections are observed in cocktail party situations where a lot people are talking in the background or in situations with an active TV or radio. We stored the speech segments and the corresponding MEL spectrograms when a keyword was correctly or erroneously detected. Using these additional data for retraining the network, we could show that the recognition performance can be steadily increased by including more and more recorded data from real life scenarios (Hirsch et al., 2020).

3.4. Energy Efficient Client

So far, we applied PI computers as the basic component for the client systems. We performed a study to find out whether a client can be operated over a period of several months with extremely low energy resources. A setup with two standard batteries of type AAA was taken as energy source. Each battery offers a voltage of 1.5 V and has an energy capacity of about 1.8 Wh. PI-based systems can be operated for about an hour given the energy resource of 3.6 Wh from two batteries and assuming an energy consumption of about 3–5 W by the PI. The PI-Zero consumes about 1 W, so that it can be operated for a few hours. Obviously, other solutions are needed at this point. This study aims at the development of a concept including hardware components and algorithmic approaches with their implementation as software to reach the goal of an extremely low energy consumption.

A lot of microcontroller units (MCU) exist that have been optimized with respect to energy consumption. We looked at a MCU with an ARM Cortex-M4 processor (Microchip, 2020a) as an example for such a device. The MCU can be operated at a voltage of less than 2 V. It can run in different modes including a special power mode where the current is dependent on the clock frequency at which the MCU runs. The value for the current is specified as 65 μA per Mhz by the manufacturer. The highest clock frequency of the MCU device is 120 MHz. Even if we would be able to realize the permanent listening for detecting the wake-up word with an efficient algorithm that could run at a clock frequency of just a few MHz, the energy consumption would be too high for operating the client over a longer period of, e.g., a few months. We started thinking about a separate analog circuit that allows the wake-up of the MCU only if the sound level exceeds a certain threshold. During our investigations of already existing solutions, we came up with the special microphone VM3011 (Vesper, 2021). This MEMS microphone and its predecessor model VM1010 have different operating modes. The VM3011 can deliver digital sample data to an MCU in its normal operating mode due to an integrated ADC. Besides this, the microphone can operate at a so-called zero-power listening mode, in which only the sound level is determined. After exceeding a certain sound level threshold the microphone sets a digital output to wake up the MCU. Then, the MCU can initiate the mode switching at the microphone. Optionally, a filtering can be enabled during the determination of the sound level. Thus, the level estimation can be focused on the frequency range of speech approximately. The sound level detection and the mode switching can be executed in the short period of a few milliseconds, so that only a short segment at the speech onset is lost. Furthermore, the VM3011 contains the feature of adapting the sound level threshold to the sound scenario in its environment. Thus, the threshold will be automatically increased in the presence of stationary background noise.

The remarkable feature of the microphone is a current of just 10 μA when running in the zero power listening mode. This allows the permanent operation over an extremely long period. Based on this microphone, we developed a processing scheme consisting of several processing stages with increasing power consumption. The processing aims at the detection and recognition of a wake-up word and the recognition of a command word or command phrase in a succeeding phase, e.g., as input for the smart home control.

As first stage, we apply a VAD algorithm to detect the beginning of speech (Hirsch and Ehrlicher, 1995). The speech signal is sampled at 10 kHz. This algorithm is based on a rough spectral analysis in 15 subbands in the frequency range from about 300 Hz to 5 kHz. The spectral analysis is performed on frames of 128 samples by applying a DFT of a length 32 on the sum of the four accumulated subframes with length 32. The energy of the background noise is estimated in each subband every 12.8 ms by looking at the smoothed energy contour. This estimation is used to define and adapt an energy threshold. In case of exceeding this threshold, the subband is considered as active. If a predefined number of active subbands is detected, the corresponding frame is considered as speech frame. The beginning of speech is indicated when speech has been detected in several consecutive frames over a period of about 100 ms. Based on the count of the needed processor cycles, we can determine a clock frequency of less than 1 MHz to run the very efficient algorithm on the MCU. We take into account the usage of the CMSLIB library (Lorenser, 2016) that has been developed for this type of ARM Cortex-M4 processor including a floating point unit. The library contains software modules for different signal processing algorithms. Detailed information is available about the number of processor cycles to realize, for example, a DFT with this library. The first processing stage can be realized with an extremely low energy consumption of the MCU. In comparison, the MEMS microphone needs a much higher current of about 700 μA in its normal operating mode.

The speech samples are buffered during the period of about 100 ms in which the beginning of speech is detected. Mel spectral features are extracted from the buffered and the succeeding samples as second processing step. Twenty-four logarithmic MEL spectral values are determined in the frequency range from about 200 Hz to 5 kHz. A DFT is applied to frames of 256 samples. The analysis window is shifted by 12.8 ms, so that we receive about 78 MEL spectra per second. We estimated again the computational resources by counting the operations that are needed for the realization of this second processing stage. The feature extraction can be implemented on the MCU at a clock frequency of <2 MHz.

As third processing stage, the feature vectors are fed into a neural network to perform either the recognition of the wake-up word or the recognition of a command phrase (Hwang et al., 2015; Sainath and Parada, 2015). The networks needed for the two tasks only differ in the number of nodes at the output layer. The structure of the network is shown in **Figure 6**.

The MEL spectra are fed into a LSTM layer with its recurrent structure to analyze and evaluate the sequence of MEL spectra. The neural network has three nodes at its fully-connected output layer for the recognition of the wake-up word. Due to a softmax scaling in the last layer, the output can be treated as the three probabilities that the sequence of MEL spectra contain the wake-up word, a speech pause or a non-keyword. The second network for the recognition of the command phrases has as many nodes at its output as the number of different commands plus one node for the speech pause and one node for the class of non-keywords. We examined the exemplary recognition of 20 German words. The 22 output values of the network for the spoken word "korrigieren" are shown in **Figure 7**. The algorithmic approach as described before has been realized with Matlab including the detection of the speech begin, the feature extraction and the recognition with the neural network. The training of the network

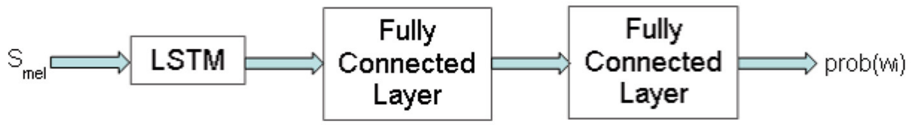

FIGURE 6 | Structure of neural network.

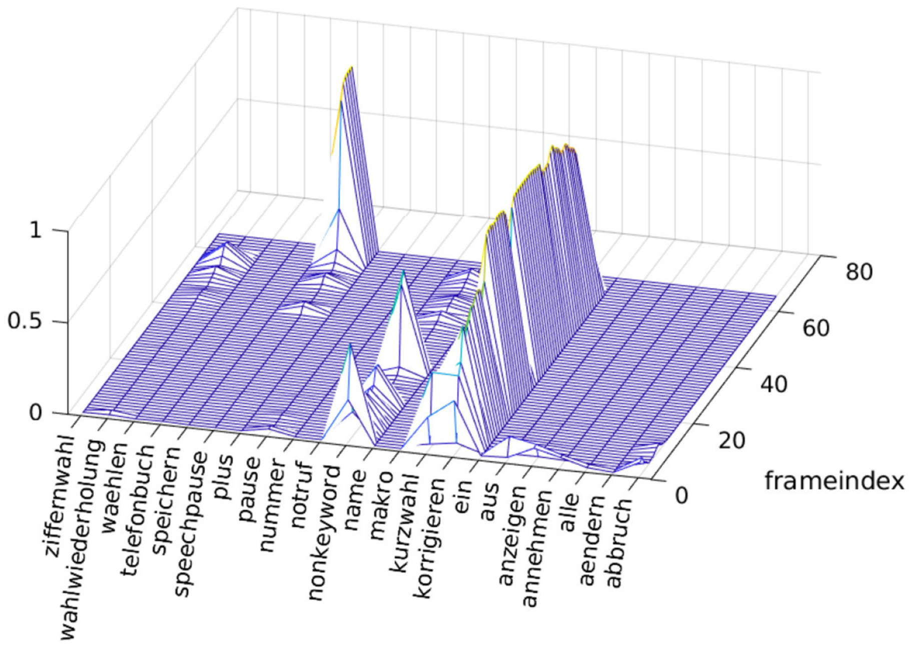

FIGURE 7 | Twenty-two output values of the neural network for the spoken word "korrigieren".

was performed with about 250 utterances of each command word and several thousand utterances of different non-keywords in Matlab.

Figure 7 shows the output values of the neural network for all speech frames after detecting the speech begin with the VAD approach described before. There are several nodes at the beginning with a noticeable value at their output. After analyzing the sequence of approximately the first 20 spectra, the output value at the node of the spoken word takes a fairly high value. Besides this node, only the non-keyword node takes some low values from time to time, but the correct recognition will be fairly easy for this example. In **Figure 7**, the criterion for detecting the end of the spoken word becomes visible. We looked at the probability at the node of the speech pause. If this probability takes on a high value in a number of consecutive frames, we take this as indication of the end of a spoken command. The final decision can be made, for example, by calculating the average probability at each node for the last 10 or 15 speech frames. Moreover, the application of a further "attention" layer is thinkable. So far, we did not deeply investigate the optimization of the recognition with respect to the feature extraction and the structure of the neural network. Clearly, our intention was the investigation whether the recognition can be realized with minimal energy consumption. We chose the small structure of the neural network with only three layers with respect to the realization on the MCU. The number of hidden units is set to 100 for the LSTM layer, the number of nodes to 200 for the first fully connected layer. Assuming a number of 22 output nodes for the recognition of 20 command words, the number of multiplications can be estimated to be at about 75,000 to realize the matrix multiplications of the three layers. Taking into account the further effort for realizing the activation functions and the calculation of the whole network at a frame rate of 78 Hz, the number of multiplications can be estimated to be at about 6.25 million per second. We derived a factor of about 8–10 from Lorenser (2016) to estimate the number of MCU cycles based on the number of multiplications. In Lorenser (2016), the number of processor cycles is listed for different signal processing tasks for which the number of multiplications is known. Thus, we can estimate that we should be able to realize the neural network with the MCU running at a clock frequency of about 60 MHz. Assuming the usage of a processor with a maximum clock frequency between 80 and 120 MHz, the computational performance would be sufficient to apply a neural network slightly more complex. There are software development frameworks for this type of MCUs (Microchip, 2020b) to create executable software modules for the integration of neural networks which were trained with tools like Tensorflow (Abadi et al., 2015).

Finally, we looked at the total energy consumption when using the microphone with the zero power listening mode in combination with an energy efficient MCU and applying the algorithmic approach as described before. The microphone would need 0.175 Wh of energy for operating in its listening mode at a voltage of 2 V for 1 year. Assuming 50 voice activations per day with the MCU running at 100 MHz for 10 s, we can estimate 0.81 Wh of energy to operate the recognition device for one year with a total current of 8 mA at a voltage of 2 V in its active phases. We have to consider that the client will also consume energy for other interfaces, e.g., for wireless communication, but it seems to be possible that such a device can be operated with minimal energy resources for a period of several months.

3.5. Communication *via* DECT-ULE

Communication between a client and a server system *via* WLAN can be problematic, for example, when different wireless networks are active in the same frequency band, e.g., at 2.4 MHz. This can delay and distort the communication process between clients and the recognition or the smart home server. We investigated the alternative use of the DECT-ULE standard for the realization of communication in the frequency band at about 1.9 GHz. DECT is used for cordless telephony in the range of up to 50 m at a fairly high transmission power of 250 mW. The ULE extension has been introduced with the goal of reducing energy consumption. Furthermore, it allows the transmission of data in addition to the speech signal. Therefore, it is suitable in the field of smart home control. With these features, it is well-suitable for communication between client and recognition server within a building. We developed a circuit based on the DHAN-M module (DSPGroup, 2020) into which the DECT-ULE protocol has been implemented. This module contains interfaces for audio input and output. Moreover, an MCU is part of our circuit as control unit of the client. The layout of the circuit has been designed to fit in an usual outlet socket. We could successfully prove the speech and data transmission between the client and a DECT base station. As the result of this short study, we consider the use of the DECT-ULE standard as an interesting alternative for communication between clients and the recognition server within a building.

4. RECOGNITION SERVER

Our goal is the realization of speech recognition in a local server system. Users can be sure that no speech data leave their homes and that nobody from outside can get access to the microphones of the clients as long as their local networks fulfill the appropriate security guidelines. In consequence, we have to aim at low cost server systems that customers are willing to pay for. This requirement is partly contrary to the need of a high computing power. We looked at three systems differing with regard to the complexity of the recognition task in the individual application. In most applications in the field of smart home control, only small vocabulary containing a few dozens or a few hundreds of words is needed. This already includes a larger variety at uttering a certain command phrase, so that the user is not forced to utter a command with only one fixed sequence of words. A fixed grammar is applied for these recognition tasks.

As the first system, we apply a PI computer which contains an ARM processor with four kernels running at a clock frequency of 1.5 GHz as recognition server. We developed a triphone-based GMM-HMM recognition scheme within the context of earlier research (Hirsch, 2008). This recognizer enables the recognition of up to a few hundred words on a PI device in real time. The triphone models have been trained on several hundreds of hours of German speech with HTK (Young et al., 2006). To increase the robustness of the recognition, data augmentation is applied to create further versions of the speech data including the acoustic effects of recording speech in hands-free mode in noisy and reverberant environments. The acoustic features consist of 12 MEL cepstral coefficients and the logarithmic energy plus the Delta and Delta-Delta coefficients. We can run this server module also on a client, so that we can set up a stand-alone device if necessary. But in case of using a PI-Zero as basis for the client, the recognition is limited to about a dozen words which could be sufficient for a simple command recognition. The dialogue software of the client is designed to freely choose different recognition servers for the individual recognition tasks within a single dialogue. Thus, we do not need communication between the client and a server at a different location in the building in dialogue situations in which only the recognition of a few words like “yes” and “no” is necessary.

We derived a second recognition system from the first one by substituting the GMM for a deep neural network (DNN). The setup of this system is shown in **Figure 8**. Spectral analysis is performed by means of a DFT at the rate of 100 frames per second. We do not determine a set of MEL cepstral coefficients as in case of the GMM-HMM recognizer. Instead, a set of 138 logarithmic compressed DFT coefficients is used as feature vector. The set of 138 coefficients consists of the DFT coefficients within the range from 250 Hz to 3 kHz. Furthermore, the mean of two DFT coefficients is calculated within the frequency range from 3 to 4 kHz and the mean of three values within the range from 4 to 7 kHz. DFT components above 7 kHz are not used. 11 consecutive vectors are taken as input for the neural net that consists of a CNN layer, a few LSTM layers and a few fully-connected layers. The number of output nodes corresponds to the number of tied triphone HMM states, which are 4,715 in our current implementation. By applying a softmax scaling at the output layer we try to estimate the emission probabilities of all tied triphone states as output of the neural network, so that we can calculate the probabilities of the HMMs as it is done with the GMM-HMM recognizer. The training of the neural net is done with Keras (Chollet, 2015) and Tensorflow (Abadi et al., 2015) by taking the same speech data as used for the training of the GMM-HMM recognizer. The mapping of feature vectors to tied triphone states was achieved by Viterbi alignment with the GMM-HMM recognizer.

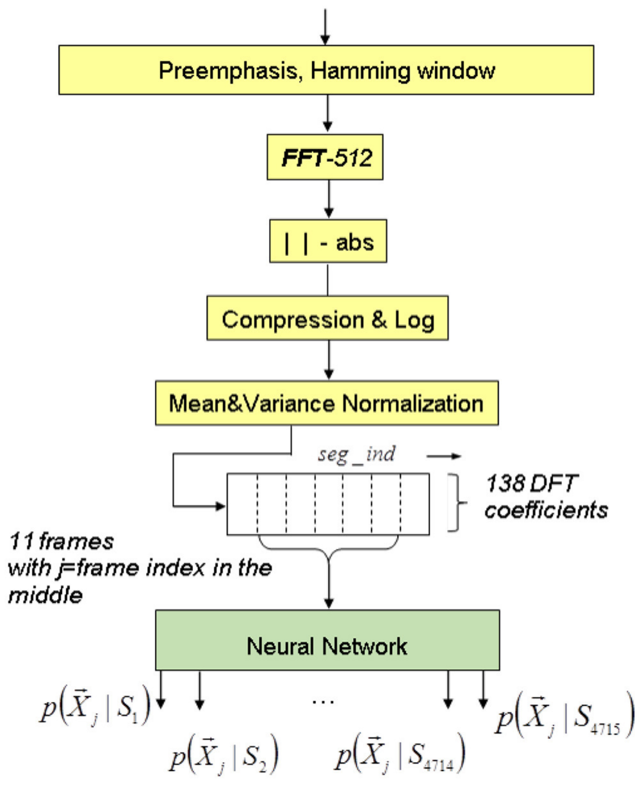
Preemphasis, Hamming window
FFT-512
| | - abs
Compression & Log
Mean&Variance Normalization
seg_ind
138 DFT coefficients
11 frames with j=frame index in the middle
Neural Network
$p(\vec{X}_j \mid S_1)$
$p(\vec{X}_j \mid S_2)$
...
$p(\vec{X}_j \mid S_{4714})$
$p(\vec{X}_j \mid S_{4715})$

FIGURE 8 | Feature extraction and determination of emission probabilities with the DNN-HMM recognizer.

TABLE 1 | Word error rates (%).

Task	GMM-HMM	DNN-HMM
Single-word	2.8	0.2
Digit-sequence	4.5	3.1
Short-sentence	27.1	15.4

The computational performance of a PI is not sufficient to realize the calculation of the emission probabilities in real time. Therefore, we take the Jetson Nano device from NVidia (NVidia, 2020) as server component. This device contains an ARM processor plus a GPU containing 128 Cuda kernels. Thus, it is well-suited for performing the matrix multiplications in order to calculate the output of the neural net. The cost of this board is higher than that of a PI, but it is affordable for usage in the field of smart home applications. The computational performance of the board is high enough to run several instances of the neural network computation in parallel. Some word error rates are listed in **Table 1** to indicate and to compare the recognition performance of the GMM-HMM and the DNN-HMM recognizers regarding three different tasks.

These error rates were achieved by training the GMMs, the HMMs, and the neural network on about 2,000 h of clean German speech data from different databases (e.g., Burger and Schiel, 1998; Radeck-Arneth et al., 2015; Pratap et al., 2020). The first task, marked as *single-word* in the table, is the recognition of a single word from a set of 64 German command words. This self-recorded database contains about 22,000 utterances in total. The second task, marked as *digit-sequence*, contains the recognition of sequences of German digits. The database consists of about 19,000 utterances with 78,500 digits in total. Moreover, the third task, marked as *short-sentence* in the table, is the recognition of the word sequence in short sentences from a train query task. The database contains about 3,200 utterances with a total of approximately 34,000 spoken words (Schiel and Baumann, 2013). No language model or grammar was applied in the third task. We allowed any sequence of words from a total of 364 words. Here, the main focus was on the word accuracy regarding a larger vocabulary. Most errors were due to the misrecognition of a declination or a conjugation that have no influence on the determination of the required information. All test data were separate sets not used for training. The performance was higher in all tasks applying the DNN rather than the GMM for determining the emission probabilities. As mentioned before, the error rates are presented to show the recognition schemes' basic performance that can be achieved in simulation experiments without the usual techniques to improve the performance in different application scenarios. Applying a client in a particularly noisy environment, we improve the recognition performance by training GMMs, HMMs, and neural networks on multi-condition data. Besides clean data, we create noisy data for the requested application scenario with appropriate tools for data augmentation (Hirsch and Finster, 2005). We have a large set of noise signals and a large collection of room impulse responses available to simulate the transmission of speech in a noisy room environment (e.g., Jeub et al., 2009; Avosound, 2022). To measure room impulse responses directly in the application scenario, we have a measuring set-up (Hirsch et al., 2010). Furthermore, we include additional garbage HMMs for modeling background noises or speech artifacts like breathing or hesitations.

In case the recognition of a larger vocabulary without a fixed grammar is needed, we looked at the Kaldi recognizer (Povey et al., 2011) as a third recognition scheme. Kaldi has become a tool that is often used for research in the field of DNN based recognition. Besides its application in the field of research, there are also versions available for the recognition in real time (Alumäe, 2014). We implemented the interface into our dialogue module on the client side to communicate with Kaldi as third recognition scheme. We train the system on the same speech data that we use for training the other recognizers. Thus, we enable our clients to have access to the recognition of a large vocabulary of ten thousands of words when it might be needed within a dialogue. The use of the Kaldi recognition can be initiated with or without a fixed grammar. Especially for the recognition without a fixed grammar, an additional module is needed for the interpretation of the recognition result. From the field of natural language processing, different methods for parsing the text string coming from the recognizer are known. Nowadays, neural networks are applied for the realization of speech interpretation. We implemented a first module into our clients based on the well known word2vec approach (Mikolov et al., 2013). But, its integration and use in the client is still an ongoing project. We implemented the real-time version of Kaldi into a low cost PC to fulfill our requirement of limited total cost.

We could proof that Kaldi can be run on such a system with almost no noticeable delay in comparison to running on a much more powerful system. Overall, we set up a system where the clients are able to freely choose one out of several recognition servers at a certain dialogue state depending on the demand of the individual recognition task at this state.

5. CONCLUSIONS

We looked at technical solutions to set up a speech assistant system that reduces the concerns of a lot of users with respect to data privacy. Our focus is on the local realization of all needed client and server components inside a local network, so that no communication in a public network is necessary to realize speech control of hardware devices in a building. Usually, the storage of the recorded speech signals or speech features is disabled on client and server side. But the user can enable the storing in his local system during an initial operating phase. This data can be used to retrain and adapt the recognition system to the acoustic environment. The local realization implies the selection or the development of cost-efficient client and server components. We presented our hardware and software approach to realize the client in a very compact shape. The client includes the dialogue control and the communication with hardware devices besides speech input and output. We conducted a separate study to realize an extremely energy-efficient client that does not need any external power supply.

Three different recognition servers have been presented. These can be applied depending on the demand and the complexity of the individual recognition task in a dialogue state. The client can easily access all servers. We could prove our concept by setting up several demonstrator systems in the field of smart home control.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary materials, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

The author confirms being the sole contributor of this work and has approved it for publication.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2015). "TensorFlow: large-scale machine learning on heterogeneous systems," in *OSDI'16: Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation* (Savannah, GA).

Alumäe, T. (2014). "Full-duplex speech-to-text system for Estonian," in *Baltic HLT 2014* (Kaunas).

Avosound (2022). *Digiffects Sound Library*. Available online at: https://www.avosound.com/ (accessed May 02, 2022).

Bäckström, T., Brüggemeier, B., and Fischer, J. (2020). "Privacy in speech interfaces," in *VDE Dialog - The Technology Magazine*. Offenbach. 11–14.

Bahmaninezhad, F., Zhang, C., and Hansen, J. H. (2018). "Convolutional neural network based speaker de-identification," in *Proceedings the Speaker and Language Recognition Workshop* (Les Sables d'Olonne), 255–260. doi: 10.21437/Odyssey.2018-36

Brasser, F., Frassetto, T., Riedhammer, K., Sadeghi, A.-R., Schneider, T., and Weinert, C. (2018). "Voiceguard: secure and private speech processing," in *Proceedings of Interspeech* (Hyderabad), 1303–1307. doi: 10.21437/Interspeech.2018-2032

Burger, S., and Schiel, F. (1998). RVG 1 - a database for regional variants of contemporary german. Available online at: https://www.phonetik.uni-muenchen.de/forschung/publikationen/Burger-98-RVG1.pdf (accessed May 02, 2022).

Chollet, F. (2015). Keras. GitHub repository. Available online at: https://github.com/fchollet/keras (accessed May 02, 2022).

Chung, H., Iorga, M., Voas, J., and Lee, S. (2017). Alexa, can I trust you? *Computer*. 50, 100–104. doi: 10.1109/MC.2017.3571053

DSPGroup (2020). *DHAN-M Module Dect Ule Platform Datasheet Version 4.0*. DSPGroup. Available online at: www.dspg.com (accessed May 02, 2022).

ETSI (2003). *Speech Processing, Transmission and Quality Aspects; Distributed Speech Recognition; Advanced Front-End Feature Extraction Algorithm; Compression Algorithm*. European Telecommunications Standards Institute document ES 202 050 v1.1.3 (2003-11).

ETSI (2019). *Digital Enhanced Cordless Telecommunications (dect)*; *Ultra Low Energy (ule)*; *Machine to Machine Communications; Part 2: Home Automation Network*. European Telecommunications Standards Institute document TS 102 939-2 V1.3.1 (2019-01).

Hernández Acosta, L., and Reinhardt, D. (2020). "Smart speakers and privacy: users? perspectives," in *VDE Dialog - The Technology Magazine* (Offenbach), 8–10.

Hirsch, H.-G. (2008). "Automatic speech recognition in adverse acoustic conditions," in *Advances in Digital Speech Transmission*, eds R. Martin, U. Heute, and C. Antweiler (Hoboken, NJ: John Wiley & Sons, Inc), 461–496. doi: 10.1002/9780470727188.ch16

Hirsch, H.-G., and Ehrlicher, C. (1995). "Noise estimation techniques for robust speech recognition," in *Proceedings of ICASSP* (Detroit, MI), 153–156. doi: 10.1109/ICASSP.1995.479387

Hirsch, H.-G., and Finster, H. (2005). "The simulation of realistic acoustic input scenarios for speech recognition systems," in *Proceedings of the 9th European Conference on Speech Communication and Technology* (Lisbon). doi: 10.21437/Interspeech.2005-263

Hirsch, H.-G., Kitzig, A., and Linhard, K. (2010). *Simulation of the Hands-Free Speech Input to Speech Recognition Systems by Measuring Room Impulse Responses*. Bochum: ITG Fachtagung Sprachkommunikation.

Hirsch, H.-G., Micheel, A., and Gref, M. (2020). "Keyword detection for the activation of speech dialogue systems," in *Proceedings Elektronische Sprachsignalverarbeitung* (Magdeburg).

Hwang, K., Lee, M., and Sung, W. (2015). *Online Keyword Spotting With a Character-Level Recurrent Neural Network*. Available online at: https://arxiv.org/pdf/1512.08903.pdf (accessed May 02, 2022).

Jeub, M., Schaefer, M., and Vary, P. (2009). "A binaural room impulse response database for the evaluation of dereverberation algorithms," in *Proceedings of International Conference on Digital Signal Processing (DSP)* (Santorini). doi: 10.1109/ICDSP.2009.5201259

Lau, J., Zimmerman, B., and Schaub, F. (2019). "Alexa, are you listening? Privacy perceptions, concerns and privacy-seeking behaviors with smart speakers privacy attitudes of smart speaker users," in *Proceedings ACM Human-Computer Interaction 2* (Glasgow, UK), 1–31. doi: 10.1145/3274371

Lorenser, T. (2016). The DSP capabilities of arm cortex-m4 and cortex-m7 processors - DSP feature set and benchmarks. Available online at: https://community.arm.com/cfs-file/__key/communityserver-blogs-components-weblogfiles/00-00-00-21-42/7563.ARM-white-paper-_2D00_-

DSP-capabilities-of-Cortex_2D00_M4-and-Cortex_2D00_M7.pdf (accessed May 02, 2022).

Malkin, N., Deatrick, J., Tong, A., Wijesekera, P., Egelman, S., and Wagner, D. (2019). "Privacy attitudes of smart speaker users," in *Proceedings on Privacy Enhancing Technologies* (Stockholm), 250–271. doi: 10.2478/popets-2019-0068

Microchip (2020a). *Data Sheet of Microcontroller Unit* Atsame53j20a. Available online at: www.st.com (accessed May 02, 2022).

Microchip (2020b). *Software Development Tool STM32CUBEIDE With the Submodules STM32CUBEMX and X-Cube-Ai*. Available online at: www.st.com (accessed May 02, 2022).

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*. Available online at: https://doi.org/10.48550/arXiv.1301.3781 (accessed May 02, 2022).

Mtbiaa, A., Petrovska-Delacrétaz, D., Boudy, J., and Hamida, A. B. (2021). "Privacy-preserving speaker verification system based on binary I-vectors," in *IET Biometrics*, eds R. Martin, U. Heute, and C. Antweiler (Hoboken, NJ: John Wiley & Sons), 233–245. doi: 10.1049/bme2.12013

Nautsch, A., Isadskiy, S., Kolberg, J., Gomez-Barrero, M., and Busch, C. (2018). "Homomorphic encryption for speaker recognition: protection of biometric templates and vendor model parameters," in *Proceedings the Speaker and Language Recognition Workshop* (Les Sables d'Olonne), 16–23. doi: 10.21437/Odyssey.2018-3

Nautsch, A., Jimenez, A., Treiber, A., Kolberg, J., Jasserand, C., Kindt, E., et al. (2019). "Preserving privacy in speaker and speech characterization," in *Computer Speech and Language: Special Issue on Speaker and Language Characterisation*, (Amsterdam: Elsevier), 441–480. doi: 10.1016/j.csl.2019.06.001

NVidia (2020). *Nvidia Jetson Nano*. Available online at: developer.nvidia.com (accessed May 02, 2022).

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., et al. (2011). "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. IEEE Signal Processing Society* (Big Island; Hawaii).

Pratap, V., Xu, Q., Sriram, A., Synnaeve, G., and Collobert, R. (2020). "MLS: a large-scale multilingual dataset for speech research," in *Proceedings of Interspeech*, 2757–2761. doi: 10.21437/Interspeech.2020-2826

Radeck-Arneth, S., Milde, B., Lange, A., Gouvea, E., Radomski, S., Muehlhaeuser, M., et al. (2015). "Open-source German distant speech recognition: corpus and acoustic model," in *Proceedings of the 18th International Conference TSD2015* (Shanghai). doi: 10.1007/978-3-319-24033-6_54

Rose, R., and Paul, D. (1990). "A hidden Markov model based keyword recognition system," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing* (Albuquerque, NM), 129–132. doi: 10.1109/ICASSP.1990.115555

Sainath, T., and Parada, C. (2015). "Convolutional neural networks for small-footprint keyword spotting," in *Proceedings of Interspeech* (Dresden), 1478–1482. doi: 10.21437/Interspeech.2015-352

Schiel, F., and Baumann, A. (2013). *Phondat 2. Bavarian Archive for Speech Signals*. Available online at: https://www.bas.uni-muenchen.de/forschung/Bas/BasPD2eng.html (accessed May 02, 2022).

Seeed (2021). *Respeaker Microphones*. Available online at: wiki.seeedstudio.com (accessed May 02, 2022).

Seiderer, A., Ritschel, H., and André, E. (2020). "Development of a privacy-by-design speech assistant providing nutrient information for German seniors," in *Proceedings of the 6th EAI International Conference on Smart Objects and Technologies for Social Good* (Antwerp). doi: 10.1145/3411170.3411227

Vesper (2021). *Data sheet of Microphone VM3011*. Available online at: vespermems.com (accessed May 02, 2022).

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Moore, X. G., et al. (2006). *The HTK Book. Version 3.4*. Available online at: https://www.academia.edu/18598764/The_HTK_book_for_HTK_version_3_4 (Accessed May 02, 2022).

Conflict of Interest: The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.