Speech Recognition, Acoustic Location and Cloud Computing in PartyBot: An IoT Device for a Party Game

Fergal Riordan
School of Computer Science & Statistics
Trinity College Dublin
Ireland
riordanf@tcd.ie

***Abstract*—This paper introduces *PartyBot*, an IoT device that uses speech recognition and cloud computing technology to implement a simple party game. In a digital age where screens often replace social contact, we hope to use modern IoT technology to work directly against this trend by introducing a game that encourages real-world, face-to-face interactions. The device uses speech recognition and acoustic location to detect a pre-defined list of *banned words* and to determine the direction from which the word was said. In addition to giving an overview of the full system, this paper will focus on the speech localisation subsystem, as well as the security and privacy of the system as a whole.**

***Index Terms*—Internet of Things (IoT), speech recognition, acoustic location, cloud computing**

I. Introduction

A. System Overview

PartyBot is a device that uses speech recognition and cloud computing to implement a party game. The concept of the game is very simple: the players define a set of *banned words* which they input to the device, then the device listens passively to the players as they converse with one another. If the device detects that one of the players has said a banned word, an alarm will sound and an arrow will point at the player who said the word.

The primary hardware component is the ESP32-s3-Korvo-2 V3.1, a comprehensive multimedia development board with a broad range of features and capabilities. The selection of this board as the primary component of the *PartyBot* system is due to its suitability for speech recognition applications. The board is built on an ESP32-S3 chip, which is the only ESP32 chip designed for SIMD (Single Input Multiple Data) instructions. This renders the chip compatible with both WakeNet and MultiNet; lightweight speech recognition models built on neural networks, specifically designed for low-power embedded MCUs. In addition to the speech recognition task, the Korvo also handles the communications with the game's user interface. These communications include users sending *banned word* updates to the device, as well as the device sending game information to the application backend on the AWS cloud computing platform for the calculation of game statistics.

While the speech recognition is implemented entirely on the Korvo using its on-board two-microphone array, the localisation of the source of each *banned word* and the actuation of the arrow to point in this direction is controlled by an ESP32-WROOM-32E device. This device is connected to four external microphones, each covering a separate quadrant of the game space. The location of each *banned word* is estimated by assuming it originated from the direction with the greatest sound intensity at the moment that the word was detected. The WROOM then uses a stepper motor to turn the arrow, pointing it in the direction of the perceived origin of the word.

B. Speech Recognition on an Embedded MCU: a Significant Dependency

The challenging task of implementing a speech recognition system on a low-power embedded MCU that can be updated dynamically via the cloud presented a significant dependency in the development process which would determine the constraints of the design of the rest of the system.

The most straightforward implementation involves performing the speech recognition on the device itself, as such a system has the simplest cloud communication requirements. Changes to the *banned word* list would simply involve the sending of text packets. However, in the early stages of the development of *PartyBot*, the group did not have access to a Korvo board and was instead working with an ESP32-S2 Kaluga. This board is not compatible with WakeNet or MultiNet and therefore an alternative method of speech recognition was originally proposed. This idea involved using a cloud-based API for speech recognition, such as the Speech-to-Text API provided by Google Cloud Platform (GCP). Had this approach been pursued, audio would be recorded by the Kaluga board's embedded microphone, and the board would then periodically encode these audio snippets and transmit them to the cloud-based speech recognition model. A message would then be returned to the board if a *banned word* was detected in the audio snippet. Had this cloud-based speech recognition been necessary, issues such as latency and security would naturally take on greater significance.

The complexity of implementing such a system on the available hardware led to an extended period of experimen-

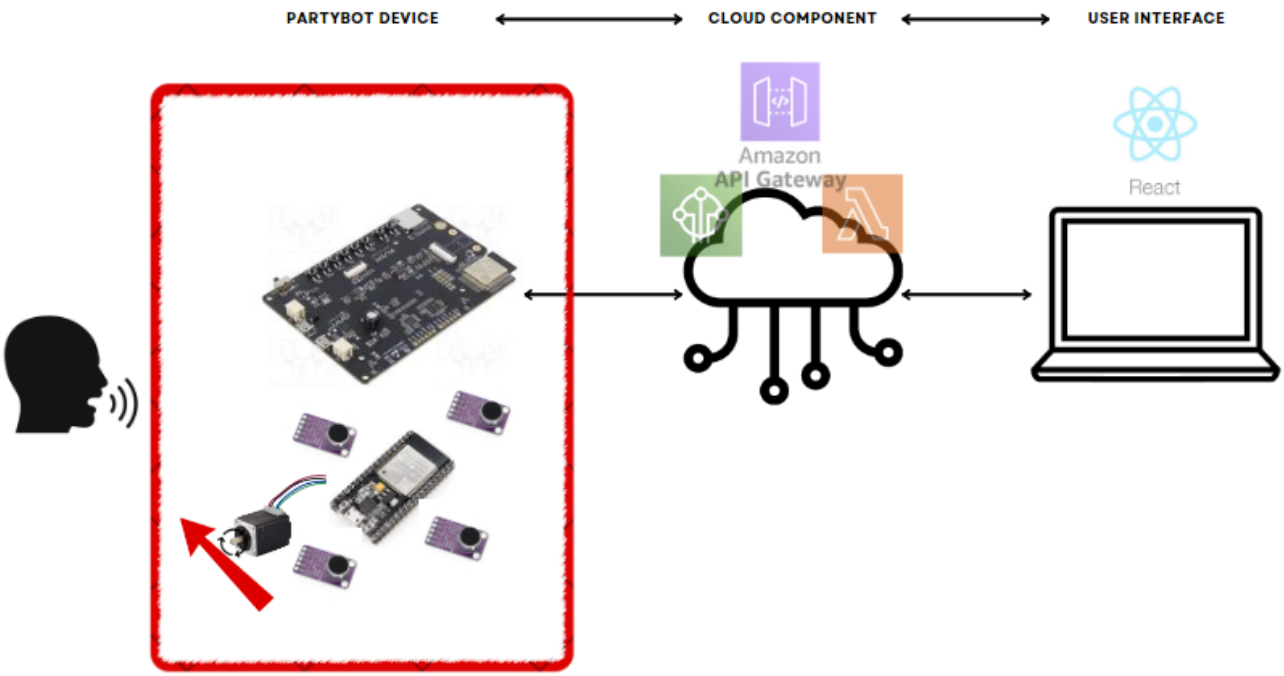

Fig. 1. The *PartyBot* system consists of a subsysem of IoT devices that interface with a cloud backend and web UI.

tation and redesigning. As the central component of the entire system, any changes to the speech recognition component could have significant effects on design choices for the rest of the system. For example, the transmission of text versus the transmission of audio packets would have major implications for the cloud component and backend logic. As a result, while the speech recognition component was not finalised, development of most of the rest of the system was suspended so that resources could be focused on overcoming this obstacle. Ultimately, with the acquisition of a Korvo board the simplest speech recognition implementation became a viable option, and thus the rest of the design was finalised. The project then transitioned from a collaborative, unified task approach to a more structured methodology, wherein the team was organized into specialized subgroups with designated responsibilities. This allowed for more focused development across different components of the system.

C. ESP Audio Development Framework

Speech recognition is implemented on the Korvo board using the Espressif Audio Development Framework (ESP-ADF). This is the official framework provided by Espressif Systems for audio application development with ESP32 devices. The framework provides comprehensive support for audio development, including the speech recognition functionality required by the *PartyBot* device. The ESP-ADF is compatible with specific versions of the Espressif IoT Development Framework (ESP-IDF); the underlying SDK for all Espressif devices.

As previously mentioned, due to the initial lack of clarity regarding the optimal speech recognition implementation and the eventual hardware change to the Korvo board, a significant amount of group work in the early weeks of this project involved the setup of and experimentation with the capabilities provided by the ESP-ADF. The ADF provides both speech recognition via the WakeNet and MultiNet models as well as general voice activity detection. During the experimentation period, both of these functionalities were implemented on the Korvo board, but ultimately only the speech recognition provided by MultiNet was deemed necessary for our purposes.

D. Cloud Component and User Interface

A key component of the *PartyBot* system is the integration of cloud computing and user interface to create an intuitive, flexible and interactive experience for users. The Amazon Web Services (AWS) cloud computing platform was selected as the foundation for this subsystem. Specifically, AWS IoTCore and AWS Lambda are used: the former for communication between the device and the cloud, and the latter for processing logic.

AWS IoTCore acts as a gateway for all communications between the *PartyBot* device and the cloud infrastructure. It provides a secure and scalable communication gateway for IoT devices to the cloud. In the *PartyBot* system, it is responsible for taking information from the device, such as the detection of a *banned word*, and forwarding this information onto AWS Lambda for processing. For communication with the Korvo,

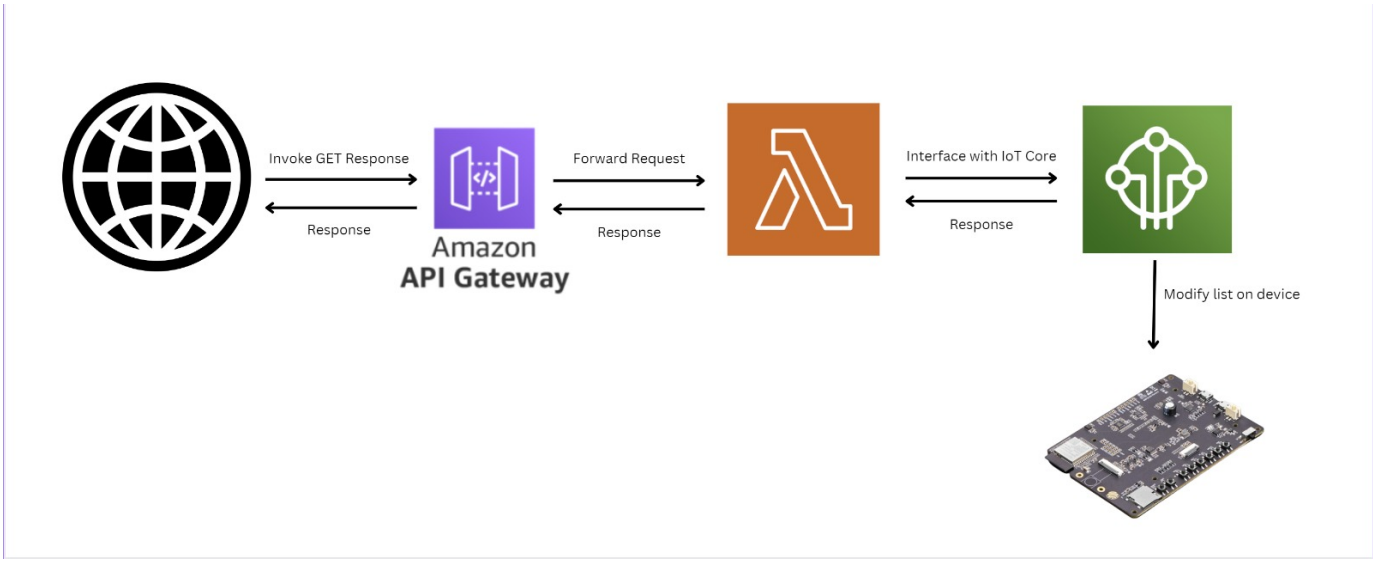

Fig. 2. Cloud component of the *PartyBot* system. The *PartyBot* device sends game statistic information to the cloud backend (AWS Lambda) via MQTT communication with AWS IoTCore. The React user interface is connected bidirectionally with the backend via the AWS API Gateway WebSocket API. This architecture facilitates seamless communication between users and the device via the UI, as well as the usage of cloud computing to remove the burden of data processing and game logic from the edge device.

the Message Queuing Telemetry Transport (MQTT) protocol is employed. MQTT is widely used in IoT applications and is perfectly suited for *PartyBot* for a variety of reasons. The protocol is designed for applications where low bandwidth usage and power consumption is of paramount importance. Furthermore, MQTT operates on a publish/subscribe model which facilitates efficient, dynamic and scalable communication between devices. On top of these primary motivating factors, the protocol's reliable message delivery, security and seamless integration with AWS IoTCore make it the obvious choice for our application.

AWS Lambda is used for data processing, such as the dynamic calculation of game statistics based on *banned word* information from the device. By outsourcing this game logic to the cloud, the computing demands placed on the edge devices is minimised, ensuring a simpler and more efficient implementation on the Korvo device as well as providing a scalable architecture that facilitates the addition of new functionality to the system, such as adding game modes or additional game metrics.

The user interface (UI) is implemented using the React JavaScript library, and provides a simple and intuitive method of interacting with the *PartyBot*. The UI interacts with cloud backend via the AWS API Gateway WebSocket API.

E. Individual Focus after defining Speech Recognition Subsystem Design

Having successfully implemented basic speech recognition functionality on the ESP32-S3 Korvo and having been able to finalise all remaining design decisions as a result, the group introduced more defined roles for team members to ensure efficient implementation of the remaining functionality. Therefore, the remainder of this paper will focus on the speech localisation, local communications (between the Korvo and WROOM), and security considerations.

II. Individual Focus: Speech Localisation, Local Communications and Security

A. Local Communication

As the *PartyBot* system involves two separate ESP32 devices - the Korvo and the WROOM - a method of local communication is required to enable the Korvo to inform the WROOM when a *banned word* has been detected. The key consideration here is latency. The speech localisation algorithm is predicated on the assumption that the message indicating the detection of a *banned word* from the speech recognition subsystem is essentially instantaneous. If the latency of the local communication is too great, the individual responsible for the *banned word* may have already stopped speaking, leading to an incorrect determination from the speech localisation system. While wireless methods were considered such as sending UDP packets over Wi-Fi, a wired option is also possible. This is therefore the obvious choice. When a *banned word* is detected by the Korvo, it will send a high signal to a digital input pin on the WROOM, which is then interpreted as an indication to immediately check for the source of the word.

B. Speech Localisation Algorithm

The subsystem for identifying the origin of a detected *banned word* is very simple. Four CMA-4544PF-W MAX9814 microphones are connected to the WROOM chip, with each microphone focusing on a separate quadrant of the space surrounding the device. The sound intensity detected by each microphone is then read to the WROOM device. In the event that the WROOM receives a message from the Korvo board

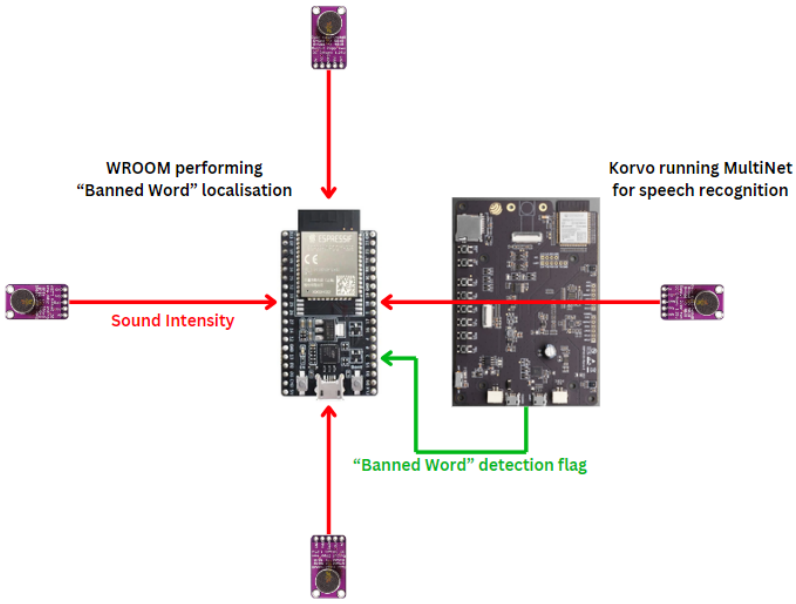

Fig. 3. The speech recognition and *banned word* localisation are performed on separate boards; the ESP32-s3-Korvo-2 V3.1 and ESP32-WROOM-32E respectively. A direct, wired method of local communication is used. While more sophisticated methods of acoustic location were considered, due to hardware constraints and project scope a simple localisation method based on sound intensity is implemented.

indicating the detection of a *banned word*, the WROOM determines which of the four microphones is currently detecting the greatest sound intensity. The direction associated with this microphone is then taken as being the source of the *banned word*.

This very basic algorithm implements a speech localisation system with four possible arrow directions. To enhance the precision of the speech localisation technique and enable the system to indicate a wider array of directions, various refinement strategies could be employed. An interpolation scheme could be implemented that also considers the sound intensity of the adjacent microphones to the microphone with the greatest intensity. Alternatively, some additional microphones could be incorporated into the design, leaving the underlying logic unchanged. The experimentation required to refine this technique has not been performed at the date of submission as the microphones were only recently acquired. However, the group's expectation is that the localisation scheme will be optimised through an iterative process of experimentation and refinement. A variety of methods of refining the intensity-based localisation technique will be explored, such as the use of a rolling average of sound intensity for each microphone rather than a single instantaneous measurement, which may be vulnerable to random noise.

Another consideration in this process of experimentation will be the latency of the system, not only from the local communication but also from the speech recognition model on the Korvo. If the delay between the statement of a *banned word* and the determination of its origin by the localisation subsystem is too great, then a mechanism will need to be incorporated into the system to account for this. One possibility will be to estimate the average delay, and simply add this delay into the maximum sound intensity logic on the WROOM. In other words, instead of programming the WROOM to take the maximum sound intensity microphone at the current time, t, as the source of the *banned word*, it instead takes the maximum sound intensity microphone of $t - delay$. The imminent experimentation process with the newly acquired microphones will reveal the viability of this solution, if it is required at all.

Finally, the speech localisation subsystem could be refined by incorporating the voice activity detection (VAD) capabilities of the ESP-ADF into the algorithm. This would mitigate the possibility of other noise sources interfering with the localisation subsystem. Experimentation with the VAD example provided in the ESP-ADF displayed very strong performance at distinguishing human speech from other noise sources, as well as distinguishing speech in the immediate vicinity of the board from background noise, even in busy environments where the background noise includes human speech. This would be a critical addition to the system in order for it to be viable commercially, as a party game will surely be primarily used in environments with numerous sources of background noise. However, due to the incompatibility of the WROOM chip with the ESP-ADF library and the time constraints of the project, this was determined to be a feature that would need to be added at a later point if *PartyBot* were to be continued as a project outside the scope of this course.

C. Security and Privacy

The security of any system that involves the passive listening of a speech recognition system to a social environment is

an obvious concern to protect the privacy of the end users. If the hardware constraints had necessitated the performance of speech recognition on the cloud instead of on the device itself, this naturally would have been a much greater concern. Fortunately, as the speech recognition is performed on the Korvo itself, the audio data recorded by the device is never stored or transmitted to the cloud: the only information regarding the user's conversations that is pushed to the cloud is an indication that a *banned word* has been detected. Additionally, as the MultiNet model is not a full-scale speech recognition model, rather a small-scale model with a recognition scope that is limited to a pre-defined set of words, there is no speech-to-text conversion of the general conversation in the room which could compromise the user's privacy in the event of a security breach. Again, as the only words the model is listening for are the pre-defined *banned words* there is no unnecessary collection of potentially sensitive information.

Security and privacy is also ensured through the use of a secure communication protocol in the form of MQTT. This protocol supports TLS/SSL for encrypted data transmission, ensuring that any data sent over the network cannot be intercepted or tampered with by third parties. Furthermore, an additional layer of security is provided through the use of a reputable and secure cloud computing platform. AWS IoTCore offers a range of advanced security features, providing mutual authentication and encryption at all points of connection.

III. Consideration of Alternative Approaches

A. Banned Word Localisation

The concept of *banned word* localisation is integral to the *PartyBot* system, and therefore warrants careful consideration of the current state-of-the-art, as well as the feasibility of the various methods for deployment in IoT devices. The *banned word* localisation task is a form of *acoustic location*; the process of determining the position of a sound source. Specifically, it is a form of *passive* acoustic location, where the localisation task relies solely upon analysis of sound waves from the object, as opposed to *active* acoustic location which involves the creation of sound to analyse the echo sound waves. Passive acoustic location was initially used primarily in military applications, specifically as a military air defense tool during World War I, but has become increasingly useful in other domains such as the detection of wildfires [1] and marine mammal localisation [2].

For *PartyBot*, the *banned word* direction is required, but the specific location of the source is not relevant. Therefore, techniques such as *triangulation*, which aim to determine the location of the source, not just the direction from which the sound originated, are not required. Instead, a simpler method such as *time difference of arrival (TDOA)* could be deployed. TDOA is based on the same principle as the human hearing system, which determines the direction of a sound source by detecting the slight time difference it takes for the sound to reach each ear. This could be applied in the *PartyBot* device with two microphones. However, due to the confined, indoor space in which the device is likely to be used, the time difference to be measured will be very small, which could affect the reliability of the system. Furthermore, with echoes, cross-talk and other sound sources in the room, the determination of the exact time of arrival of the sound at each microphone is likely to be a significant challenge. For these reasons, the TDOA method of acoustic location was not implemented, and instead the sound intensity of each microphone is used, as previously discussed.

B. Voice Activity Detection

While speech recognition is an integral element of *PartyBot*, voice activity detection (VAD) is another technique which was investigated as a potential inclusion to improve the robustness of the system. While speech recognition aims to convert speech recordings to text, VAD simply aims to determine whether speech is actually present in an audio recording. VAD is often used in conjunction with speech recognition techniques, for example to isolate speech data from background noise as a means of improving the performance of the speech recognition system [3]. An obvious application of VAD for *PartyBot* is in the *banned word* localisation. The addition of VAD would enable the device to distinguish between voice activity and other sounds, such as music, which may otherwise affect the accuracy of the system.

VAD is a rich area of research, with a wide variety of approaches. The simplest VAD systems employ an energy-based approach, where speech is detected when the sound signal energy rises a threshold amount above the noise floor. Several more sophisticated approaches to VAD have been proposed [4], but in recent years, deep learning approaches to VAD have risen to prominence [5].

VAD capabilities are included in the ESP-ADF, though the VAD algorithm is not open-source, therefore the specifics of its implementation are not known. The VAD system can be integrated with the speech recognition capabilities provided by WakeNet and MultiNet, though the only alteration that can be made to the VAD component is to alter the sensitivity threshold. Experimentation with the VAD example provided in the ESP-ADF indicated strong performance, and it was concluded that VAD would be a worthwhile inclusion in *PartyBot*. However, due to the incompatibility of the ESP32-WROOM-32E with the ESP-ADF and the limited scope of this project, the feature was not included in the *banned word* localisation scheme for the *PartyBot* prototype.

C. Security Considerations

As previously discussed, the performance of the speech recognition task locally using a limited speech recognition model that is only capable of recognising a limited, pre-defined set of words alleviates many of the privacy concerns associated with a device like *PartyBot*. However, careful consideration of the tools to ensure a completely secure system remains a key issue.

The choice of an appropriate application layer protocol is key to ensuring the security of the system, and the decision to use MQTT as the communication protocol was made with

this in mind. While several other options are available such as Constrained Application Protocol (CoAP), Advanced Message Queueing Protocol (AMQP) and Extensible Messaging and Presence Protocol (XMPP) [6], ultimately MQTT was chosen for its balance of lightweight overhead and robust security.

IV. Conclusion

Through the development of *PartyBot*, a variety of insights have been gained into the diverse set of challenges that arise in the development of IoT devices. In particular, the influence of hardware limitations on the overall design of a system was highlighted by the initial plan to use an ESP32-S2 Kaluga, the subsequent shift to the ESP32-S3 Korvo-2 V3.1, and the effect that this shift had on the cloud component of this project.

Another significant learning point was the pivotal role of communication protocols in the security and functionality of IoT devices. While MQTT was ultimately chosen as the communication protocol for *PartyBot*, the investigation of alternative protocols, and the trade-offs between security, computational overhead and other factors, highlighted the importance of thoughtful design decisions across all components of such an IoT system.

While the basic application implemented in *PartyBot* provides a great foundation for a popular party game, some refinements are required to improve the user experience and bring *PartyBot* closer to the end goal of a viable consumer product. One obvious weakness of the system in its current form is the web UI: a mobile app would be a more natural form of UI for such a device. Additionally, improving the speech localisation algorithm, perhaps by incorporating VAD, is another area for further work.

An additional consideration with the foundation laid by *PartyBot* is the potential for this device as a basic template for a variety of use cases. The use of cloud computing to offload much of the computation from the edge device provides a flexible framework for repurposing *PartyBot* with new game logic, or perhaps as a language learning tool.

References

[1] Huang H-T, Downey ARJ, Bakos JD. "Audio-Based Wildfire Detection on Embedded Systems." Electronics. 2022; 11(9):1417. https://doi.org/10.3390/electronics11091417

[2] Eva-Marie Nosal; "Methods for tracking multiple marine mammals with wide-baseline passive acoustic arrays." J. Acoust. Soc. Am. 1 September 2013; 134 (3): 2383–2392.

[3] Ghahabi, O., Zhou, W., & Fischer, V. (2018). "A robust voice activity detection for real-time automatic speech recognition." Proc. ESSV, 85-91.

[4] Jongseo Sohn, Nam Soo Kim and Wonyong Sung, "A statistical model-based voice activity detection," in IEEE Signal Processing Letters, vol. 6, no. 1, pp. 1-3, Jan. 1999, doi: 10.1109/97.736233.

[5] Korkmaz, Y., & Boyaci, A. (2023). "Hybrid voice activity detection system based on LSTM and auditory speech features." Biomed. Signal Process. Control., 80, 104408.

[6] Gerodimos, A.; Maglaras, L.; Ferrag, M.A.; Ayres, N.; Kantzavelou, I. "IoT: Communication Protocols and Security Threats." Internet of Things and Cyber-Physical Systems 2023, 3, 1–13, doi:10.1016/j.iotcps.2022.12.003.