Text-based Voice Codec Algorithm for Tactical Radio Networks in Disconnected, Intermittent, Limited Environment*

1st Jongdeog Lee
Dept. of Computer Science
Korea Military Academy
Seoul, South Korea
jdlee6461@kma.ac.kr

2nd Jongkwan Lee
Dept. of Computer Science
Korea Military Academy
Seoul, South Korea
jklee6456@kma.ac.kr

***Abstract*—Operations on the battlefield are becoming increasingly dependent on tactical networks as more robots, drones, and mobile devices are deployed. A tactical network is considered a disconnected, intermittent, limited environment, considering adversary attacks on network infrastructure. While tactical radios transmit voice messages over tactical networks, recipients may suffer message delays owing to limited bandwidth. Herein, we propose a new codec algorithm that converts voice to text using a voice recognition algorithm to deliver semantic information with minimal bandwidth consumption. We also provide an adaptive codec selection algorithm that selects the appropriate encoding algorithm according to the network capacity based on the trade-off between the compression rate and sound quality. The algorithm exploits the text-based voice codec algorithm in the minimal network environment; otherwise, it adaptively selects the best audio codec considering the network condition. Experiment results show that the proposed text-based codec can deliver messages under extremely scarce network conditions.**

***Index Terms*—voice codec, voice recognition, IoT, tactical networks**

I. INTRODUCTION

Since the introduction of network-centric warfare, networks have become essential for achieving victory on the battlefield. Its importance in future warfare, in which various sensors, drones, and robots are deployed, is expected to grow further. Although the networks used on the battlefield include both satellite and backbone networks, the network to be used at the tactical level comprises an ad-hoc method primarily composed of radios in consideration of mobility. Tactical radios are considered network nodes that configure ad-hoc networks in modern battlefields in addition to transmitting and receiving messages between military units.

The problem with tactical networks is the low bandwidth. Providing high bandwidth tends to be challenging because the frequency capable of providing a high bandwidth is used for civilian use. High-capacity multimedia data transmission is required on the battlefield because of the increased use of data-based algorithms, including machine learning. Considering the limited bandwidth, the transmission of multimedia data over tactical networks is a heavy burden. In particular, a jamming attack is difficult to defend against, which makes it a bigger problem.

Owing to these problems, the tactical network is generally classified as a disconnected, intermittent, limited (DIL) environment. Even in DIL environments, message and data sharing should be conducted faster than the enemy forces to win by attacking the enemy first through quick decision-making processes. At a minimum, ensuring smooth transmission and reception of voice messages over radio is essential. However, the amount of data to be shared, the enemy's jamming attacks, and voice message transmission can also be challenging, considering the limited bandwidth of tactical networks.

This study aims to ensure reliable communications between radios in a tactical network in DIL environments. Despite having a significantly smaller capacity than video data, audio data representing human voice is still multimedia data that can be a burden in a limited network environment. Therefore, we propose a new voice encoding algorithm that converts voice to text using a voice recognition algorithm. Then, the

This work was supported by the Hwarang-dae Research Institute of Korea Military Academy and by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIP) (No. 2020R1G1A1005288135648200077010 3)

recipient restores the sender's voice by playing an adaptive codec selection algorithm that can transmit data within the required time by increasing the compression rate even if the sound quality is sacrificed to some degree according to the network capacity. This algorithm exploits the trade-off between the sound quality and compression rate to transmit high-quality data if the network bandwidth is sufficient and transmits data with a high compression rate otherwise. In particular, when transmitting voice data is difficult, we propose converting audio data into text data using a voice recognition algorithm and transmitting the converted text through a network. Text data requires tens of bytes at most when expressing one sentence, which is a size that can be transmitted even in a limited network.

The advantages of adaptively converting voice data are reduced network bandwidth usage and message delay. More messages can be shared because the data to be transmitted is significantly reduced; hence, message latency is also reduced. Subsequent packets are consecutively delayed for real-time voice messages once a packet is delayed because of the bandwidth reduction. Therefore, the network congestion may be reduced while resolving the inconvenience experienced by the user by adaptively converting a message with a high compression ratio. In particular, this guarantees the transmission of semantic data even in a resource-limited network environment.

A scenario in which the bandwidth decreases sequentially was generated to confirm that the adaptive codec selection algorithm converts data types flexibly according to changes in the external network. In this scenario, the bandwidth is reduced to half with time and eventually returns to the original. The adaptive codec selection algorithm determines network congestion through the end-to-end delay time. The algorithm selects the codec with the highest sound quality when sufficient network bandwidth is secured. Otherwise, a codec with low sound quality, but a high compression rate is selected. When a delay occurs in all voice codecs, data converted into text are transmitted to reduce the transmission delay and solve the network congestion.

The contributions of this study are threefold. First, an adaptive codec selection algorithm is developed. In DIL environments, voice data are converted to fit the network capacity by leveraging the trade-off between the sound quality and compression rate. Accordingly, the message is delivered within the time required by the application, guaranteeing seamless transmission and providing a maximum sound quality within a range in which transmission delay does not occur. Second, a speech-to-text (STT) technology is used for network congestion control. Network resources are limited and difficult to expand dramatically. In particular, this problem is even more severe in environments such as tactical networks. This study uses STT computation to operate limited network resources effectively; that is, when the bandwidth is limited, the load of the network is reduced using the STT operation. Although several attempts have been made to apply STT technology to military applications; to the best of our knowledge, no use for saving bandwidth in radio networks has been developed. Finally, a quantitative analysis based on simulations is conducted. The effectiveness of the adaptive codec selection algorithm, including the voice codec and recognition algorithms, is demonstrated to prove their feasibility. In addition, the results concerning bandwidth consumption and delay time can be referenced in future studies.

The remainder of this study is structured as follows. Section II discusses existing voice codecs, voice recognition algorithms, and their applications in the military field. Section III presents the concept of an adaptive codec selection algorithm and discusses its design and implementation. Section IV analyzes the performance of existing speech codecs and speech recognition algorithms and verifies whether the adaptive codec selection algorithm responds to external network changes as expected. Finally, Section V concludes this work.

II. Related work

A voice codec is a core technology necessarily used in applications that transmit the human voice. Pulse-code modulation (PCM) is the most basic digital conversion technique to digitally express a sampled analog signal through sampling, compression, quantization, and encoding processes. A- and μ-law algorithms belong to this category. Linear prediction is a speech coding technique developed and widely used since 1990s; most modern standards are based on linear prediction [1], [2]. An early speech codec using a linear prediction formula is differential pulse code modulation (DPCM). DPCM is used in ITU-T standards G.721, G.726, and G727 and has speeds of 16 to 40 Kbps. ADPCM does not use the predicted value of DPCM as a constant but dynamically determines the predicted value according to the sample and encodes the difference between the predicted values to increase the compression rate [3].

Code-excited linear prediction (CELP)has been the dominant speech codec since 2000s [4], [5]. ACELP, one of its variations, is the G.729 standard and provides an 8-Kbps speed [6]. The adaptive multi-rate (AMR) codec uses the ACELP method but can offer different speeds using sub-band vector quantization [7].

In the field of natural language processing, automatic speech recognition (ASR) technology enables a computer to understand and execute human speech [8], [9]. The technical difficulties of ASR are primarily in signal processing and feature extraction because of the differences in pronunciation and accent among individuals. This problem is exacerbated when considering factors such as gender, social tendency, conversational style, and the 6500 languages spoken worldwide [10], [11]. Previously, the system accuracy was increased through a "training" period in which the user reads sentences or words, and the system understands them. However, massive vocabularies have recently been used to study systems that are independent of the speaker, eliminating the need for such training periods [12]. Furthermore, since the development of deep learning technology in 2010, the performance of ASR has improved dramatically [13]–[17].

Voice codecs and recognition technologies are already utilized in military fields. Voice codecs are a necessary technology for military radio communications, and various voice codecs are used according to specific purposes. The AMR codec is similar to this study in that it adjusts the speed according to the external environment. However, the difference is that while AMR adjusts the speed through commands from the mobile switch center (MSC) or the eNodeB of LTE, this study proposes adjusting the speed through the feedback control of each node according to the external environment. Additionally, although voice codecs have high compression rates, they still require a transmission speed in Kbps. Conversely, this study enables message transmission with a speed in bps via text conversion. Hence, the proposed method can provide the minimum functionality even in extreme environments.

III. Protocol design

This section describes the proposed adaptive message selection algorithm and its implementation.

A. Algorithm Design

Tactical radios use one voice codec because multiple codecs complicate the software implementation. However, we used multiple voice codecs to leverage their benefits, particularly in supporting the appropriate codec based on the network environment. Typically, the compression ratio and sound quality are in a trade-off relationship; that is, the sound quality deteriorates if the compression ratio is high. Conversely, the sound quality improves if the compression ratio is low.

Therefore, if the network is deemed congested, the compression ratio can be increased to reduce the delay time by measuring the end-to-end delay time in the application layer of the network. Moreover, if the network is not congested, the sound quality can be increased to improve the service quality.

Suppose the network bandwidth is highly restricted because of adversary attacks such as jamming and thus becomes extremely congested. Even the voice codec with the highest compression ratio cannot transmit messages in time because at least kilobytes of data are necessary for transmitting voice. However, text can exchange data only in bytes such that it can be transmitted even in highly congested environments. Therefore, we can overcome this limitation by converting voice into text.

The end-to-end delay time of the network must first be measured to implement the adaptive message selection algorithm described above. Hence, the sender sends the voice message data along with the current timestamp. The client measures the time of receipt to calculate the elapsed time for the one-way trip, assuming that clock synchronization between sender and receiver is established. Several methods can synchronize the time; however, in this study, time synchronization is not a focus. Therefore, two end nodes are assumed to be time-synced.

For tactical radios, a threshold is set before manufacturing to prevent delays from exceeding a specific time. If the measured delay exceeds the threshold, the network bandwidth is assumed to be insufficient, and the transmitter is signaled to select a message type with a high compression ratio. Evidently, temporary transmission delays may occur owing to instability; thus, instead of reacting immediately, waiting for a certain amount of time is preferable. If the data transmission rate changes quickly, the data flow in the network may become unstable and difficult to predict.

We can design an algorithm to change the message delivery method with patience in temporary and exceptional situations by setting variables such as a buffer. Algorithm 1 details such an algorithm. T is the threshold of the message delay time, l is the initial message level, and L is the maximum level

of messages. Furthermore, p denotes the patience, a variable that acts as a buffer, and P denotes the maximum size of the buffer.

Algorithm 1: Adaptive Message Selection

```
T ← threshold;
l ← initial message level;
L ← maximum message level;
p ← initial patience;
P ← maximum patience;
while TRUE do
    d ← RTT delay;
    if d ≥ T then
        p ← p/2 ; /* Half Patience */
        if p ≤ 1 then
            l ← Min(1, l − 1) ;
            /* Decrement Message
            Level if l > 1 */
        end
    else
        p ← p + 1 ;        /* Increment
         Patience */
        if p ≥ P then
            l ← Max(L, l + 1) ;
            /* Increment Message
            Level if l < L */
        end
    end
end
```

The one-way delay (d) of a message is measured for every transmission, and if d is greater than or equal to T, then the patience (p) is halved. Otherwise, p is increased by 1. This mechanism is the same as the additive increase/multiplicative decrease (AIMD) in TCP congestion control, which adjusts the window size according to the network environment. The AIMD mechanism increments the window size when no network congestion is observed and halves it when congestion is observed to minimize network overflow. Herein, the adaptive message selection algorithm adjusts p according to the network congestion using the AIMD mechanism to quickly determine the appropriate message level.

If p continuously decreases and reaches 1, then the message level is lowered. Subsequently, a high-quality and low-compression codec is changed to a low-quality and high-compression codec to transmit messages even in congested environments. Suppose the network returns to normal; p increases and exceeds P. The algorithm then selects high-quality and low-compression methods again to meet service quality.

B. Implementation

A simple message server/client program is programmed in Python to implement the proposed algorithm. This program receives sound from a microphone and sends it to the client through a message server. The ZeroMQ [18] library was used to construct the server and client for sending and receiving messages.

ZeroMQ supports sockets and various messaging patterns, including Req/Rep and Pub/Sub patterns. The Req/Rep pattern enables clients to request messages from the server, which then analyzes the message and responds accordingly. The Pub/Sub pattern is a subscription service, in which subscribers can automatically receive messages published by the publisher by subscribing to messages they want to receive.

This study exploits both message patterns (Pub/Sub and Req/Rep) between the sender and the receiver. The sender asks the receiver what message level should be used before transmitting the message. The receiver replies with the message level set to the default value if it is the first message. The receiver selects an appropriate message level according to the algorithm described if a previous message record exists.

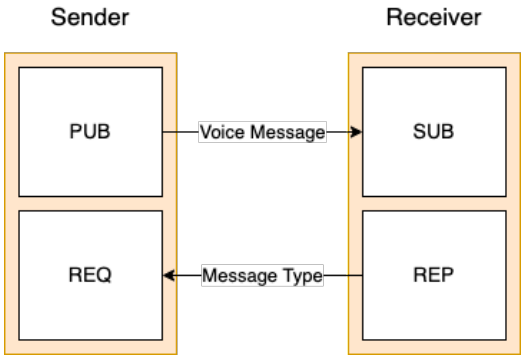

Fig. 1. Message pattern between sender and receiver.

Once a consensus is reached on the message level, the sender selects a codec matching the level and transmits the voice message to the recipient using the Pub/Sub method. A high-quality and low-compression codec is selected for a network environment with sufficient bandwidth. Conversely, a low-quality and high-compression codec is selected for insufficient bandwidth. For extreme bandwidth

restrictions, text data converted by voice recognition are transmitted. Figure 1 shows the Pub/Sub pattern of the sender and receiver as well as the Req/Rep pattern.

This study used A-law, μ-law, and ADPCM methods supported by the audioop library [19] as audio codecs. The following section details each algorithm. The ADPCM method has a higher compression ratio than that of the PCM method of the A- and μ-law algorithms; thus, the voice service is initially provided using the A- or μ-law method. If a message delay is detected, the algorithm selects ADPCM. The final solution is to suggest converting voice data into text for a minimal network bandwidth.

Several libraries provide speech recognition functionality; however, some operate using a cloud-based approach, which requires processing speech data to remote servers. The cloud-based approach does not align with the objectives of this study, which aim to minimize network bandwidth usage; therefore, PocketSphinx [20], a library that can be installed locally and used, was selected.

IV. Simulation

This section analyzes the performance of codecs and voice recognition algorithms used in the adaptive message selection protocol.

A. Dataset

The data used for this experiment comprises the "The LJ Speech Dataset" shared on Kaggle [21]. This dataset consists of 13,100 short audio clips of a single speaker reading seven books, and scripts for each clip are provided. The length of each clip varies between 1 and 10 s, and the total length is approximately 24 h. In the experiment, the audio clips are assumed to be voice messages sent from the sender to the recipient.

The LJ Speech Dataset features speakers with stable pronunciation and speed, free of background noise. However, in real-world combat scenarios, the audio data is often contaminated by noise, such as gunshots and explosions, making speech recognition challenging. A dataset of audio data from actual combat operations is available [22]; however, its poor sound is difficult for even human listeners to comprehend. Furthermore, the audio clips stored in the dataset likely suffer from significant data loss compared with the original audio, making them poor to use as representative of actual conditions in real-world scenarios. Despite this, the poor recognition rate observed when applying speech recognition algorithms to this dataset does not necessarily imply that such algorithms are unusable in tactical situations but rather suggests the need for pre-processing steps such as noise reduction.

The LJ Speech Dataset may not be considered speech data in a tactical situation. However, it can be assumed to be data after preprocessing, such as noise removal, is performed. This is left for future research because techniques for preprocessing speech data in tactical situations do not exactly match the main topic of this study. The experimental data values described later can be understood as experimental results for audio data in which such preprocessing was well performed.

B. Scenario

The adaptive message selection algorithm uses high-quality codecs when the bandwidth is sufficient and switches to low-quality codecs as the bandwidth decreases. For any speech codec restricting transmission, we propose a method that converts speech into text to convey at least a minimal amount of semantic information. This can be changed back to a high-quality codec when the network environment improves. Section III details the adaptive message selection algorithm.

TABLE I
MESSAGE LEVELS IN THE ADAPTIVE MESSAGE SELECTION ALGORITHM

Level	Message Type	Compression	Sound Quality
4	RAW	Very Low	Very High
3	A-law PCM	Low	High
2	ADPCM	High	Low
1	TEXT	Very High	-

As shown in Table I, four message types are defined for the adaptive message selection algorithm to choose from. First, "RAW," which has the highest quality codec, directly transmits the data received from the microphone such as the WAV format. Second, the A-law PCM encodes using the A-law algorithm and then transmits it. Performance-wise, no difference is observed between the A- and μ-law algorithms; thus, only the A-law algorithm was used. Third, ADPCM, which has a higher compression rate but a slightly degraded sound quality, is defined as the second-level message type. Finally, when the network environment is limited, the "TEXT" transmits converted text using speech recognition algorithms.

A scenario was created, as described in Table II, to investigate how the adaptive message selection algorithm adapts to changes in the network bandwidth. The scenario is 6-min long, in which messages are transmitted without bandwidth limitations during the first minute. During the next minute, the bandwidth is restricted to 128 Kbps. Subsequently, then on, the bandwidth was halved every minute. The last minute was messages are transmitted at the original bandwidth of 1 Gbps.

TABLE II
CHANGES IN NETWORK BANDWIDTH DURING THE SCENARIO RUN

Time (s)	Bandwidth	Time (s)	Bandwidth
0-60	1 Gbps	180-240	32 Kbps
60-120	128 Kbps	240-300	16 Kbps
120-180	64 Kbps	300-360	1 Gbps

C. Results

We measured the bandwidth usage and message delay of the adaptive message selection algorithm and ADPCM, the most efficient voice codec algorithm, using the aforementioned scenario. Figure 2 shows the bandwidth usage. For ADPCM, even when the bandwidth is sufficient for the first 3 min, it uses only a bandwidth of approximately 50 Kbps. Then, it is unable to use the bandwidth adequately, which manifests as delayed messages. For the last minute, as the bandwidth is restored, delayed messages are delivered instantly, resulting in a rapid increase in bandwidth usage.

Furthermore, the adaptive message selection algorithm initially uses a bandwidth of approximately 350 Kbps for the first minute, then switches to A-law PCM with a bandwidth of approximately 100 Kbps when the bandwidth decreases to 128 Kbps. When it changes to 64 Kbps, the algorithm switches to ADPCM, using a bandwidth of approximately 50 Kbps. From the logs, some delays are noted when the ADPCM data reaches 32 Kbps; however, it does not exceed the 2-s threshold; thus, the ADPCM messages continue to be transmitted. At the 5- min mark, when the bandwidth is changed to 16 Kbps, the algorithm completely switches to TEXT mode, transmitting messages without consuming nearly any bandwidth. Data are transmitted at high speeds after the original bandwidth recovers, raising the level to the RAW mode.

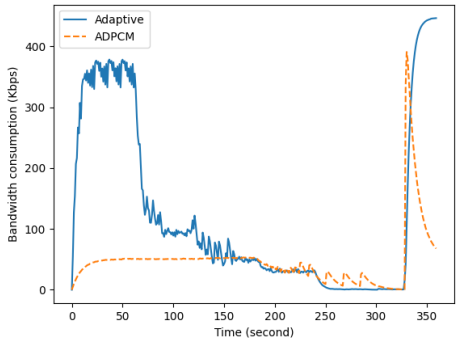

Fig. 2. Bandwidth consumption of adaptive conversion algorithm (Adaptive) and ADPCM according to the bandwidth changes

Figure 3 illustrates a graph indicating the departure time (x-axis) and arrival time (y-axis). For the ADPCM algorithm, messages are transmitted almost without delay in environments with bandwidths above 64 Kbps. However, after 180 s, the messages continue to be delayed, and the arrival time rapidly increases. Two peaks are observed for the adaptive algorithm. The first is when the bandwidth decreases from 1 Gbps to 128 Kbps, and consequently, the message type is changed when the delay time exceeds the threshold (2 s). Then, the message type is changed to ADPCM as the threshold is exceeded again when the network speed of 64 Kbps decreases again. Subsequently, the delay time approaches the threshold near the 32-Kbps bandwidth and ultimately converts to text mode at 16 Kbps, transmitting messages very quickly.

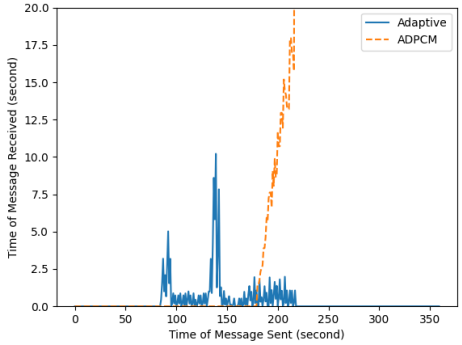

Fig. 3. Comparison of the transmission speed of voice and text data

This experiment verifies that the adaptive message selection algorithm can convert the message type

to match the bandwidth. Further research on setting optimal parameters, such as the delay time threshold T and parameter p for the adaptive message selection algorithm, should be conducted. Depending on these values, the algorithm can respond more sensitively or insensitively to external changes and dynamically change the values by determining whether the problem is temporary or permanent. Further research on dynamically setting the parameters of the adaptive transformation algorithm is left for future research.

V. Conclusion

This study proposed an adaptive message selection algorithm for effectively transmitting voice messages from tactical radios under limited bandwidth conditions. The network bandwidth was determined through the end-to-end message delay time, and if the bandwidth was sufficient, high-quality and low-compression voice codecs were used. Conversely, low-quality and high-compression voice codecs were used if the bandwidth was insufficient. We aim to provide uninterrupted service in minimal network environments even when using high-compression voice codecs if transmission delays occur by converting voice data into text using voice recognition algorithms.

This paper does not discuss setting the parameter (T, p, P) values of the adaptive message selection algorithm. However, adaptively optimizing the values according to the network environment or application needs is an interesting research direction for future work. In future, several other strategies in addition to the AIMD algorithm can be used to minimize the network overflow.

References

[1] J.-H. Chen and J. Thyssen, *Analysis-by-Synthesis Speech Coding*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 351–392. [Online]. Available: https://doi.org/10.1007/978-3-540-49127-9_17

[2] D. J. Sinder, I. Varga, V. Krishnan, V. Rajendran, and S. Villette, *Recent Speech Coding Technologies and Standards*. New York, NY: Springer New York, 2015, pp. 75–109. [Online]. Available: https://doi.org/10.1007/978-1-4939-1456-2_4

[3] K. C. Pohlmann, *Principles of Digital Audio*, 4th ed. McGraw-Hill Professional, 2000.

[4] K. Sayood, *Introduction to Data Compression, Fourth Edition*, 4th ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2012.

[5] R. Jage and S. Upadhya, "Celp and melp speech coding techniques," in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2016, pp. 1398–1402.

[6] R. Salami, C. Laflamme, J.-P. Adoul, A. Kataoka, S. Hayashi, T. Moriya, C. Lamblin, D. Massaloux, S. Proust, P. Kroon, and Y. Shoham, "Design and description of cs-acelp: a toll quality 8 kb/s speech coder," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 2, pp. 116–130, 1998.

[7] T. Ogunfunmi, R. Togneri, and M. S. Narasimha, *Speech and Audio Processing for Coding, Enhancement and Recognition*, 1st ed. Springer Publishing Company, Incorporated, 2016.

[8] S. Saksamudre, P. Shrishrimal, and R. Deshmukh, "A review on different approaches for speech recognition system," *International Journal of Computer Applications*, vol. 115, pp. 23–28, 04 2015.

[9] A. Y. Vadwala, K. A. Suthar, Y. A. Karmakar, and N. Pandya, "Survey paper on different speech recognition algorithm: Challenges and techniques," *International Journal of Computer Applications*, vol. 175, pp. 31–36, 2017.

[10] M. Forsberg, "Why is speech recognition difficult," 03 2003.

[11] D. O'Shaughnessy, "Invited paper: Automatic speech recognition: History, methods and challenges," *Pattern Recogn.*, vol. 41, no. 10, p. 2965–2979, oct 2008. [Online]. Available: https://doi.org/10.1016/j.patcog.2008.05.008

[12] M. Malik, M. K. Malik, K. Mehmood, and I. Makhdoom, "Automatic speech recognition: A survey," *Multimedia Tools Appl.*, vol. 80, no. 6, p. 9411–9457, mar 2021. [Online]. Available: https://doi.org/10.1007/s11042-020-10073-7

[13] M. Shahin, B. Ahmed, J. Mckechnie, K. Ballard, and R. Gutierrez-Osuna, "A comparison of gmm-hmm and dnn-hmm based pronunciation verification techniques for use in the assessment of childhood apraxia of speech," 09 2014.

[14] M. Y. Tachbelie, A. Abulimiti, S. T. Abate, and T. Schultz, "Dnn-based speech recognition for globalphone languages," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8269–8273.

[15] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[16] A. Rista and A. Kadriu, "Automatic speech recognition: A comprehensive survey," *SEEU Review*, vol. 15, pp. 86–112, 12 2020.

[17] R. H. Sun and R. J. Chol, "Subspace gaussian mixture based language modeling for large vocabulary continuous speech recognition," *Speech Commun.*, vol. 117, no. C, p. 21–27, feb 2020. [Online]. Available: https://doi.org/10.1016/j.specom.2020.01.001

[18] "Zeromq." [Online]. Available: https://zeromq.org/

[19] "Audioop." [Online]. Available: https://docs.python.org/3/library/audioop.html

[20] "Pocketsphinx." [Online]. Available: https://pypi.org/project/pocketsphinx/

[21] "The lj speech dataset." [Online]. Available: https://www.kaggle.com/datasets/mathurinache/the-lj-speech-dataset

[22] "Military audio clips." [Online]. Available: https://www.radioheritage.com/Shortwave-With-A-Difference/milaudio.htm