İSTANBUL UNIVERSITY CERRAHPAŞA

ORIGINAL ARTICLE
Electrica 2025; 25: 1-9

Artificial Intelligence Meets Your Voice: Transforming Turkish Text into Personalized Speech

Funda Akar

Department of Computer Engineering, Erzincan Binali Yıldırım University, Faculty of Engineering and Architecture, Erzincan, Türkiye

Cite this article as: F. Akar, "AI meets your voice: transforming Turkish text into personalized speech," *Electrica*, 25, 0034, 2025. doi: 10.5152/electrica.2025.25034.

ABSTRACT

Advancements in artificial intelligence (AI)-driven text-to-speech (TTS) technology have enabled transformative applications in accessibility and personalized learning. This study focuses on the development of a customizable Turkish TTS system aimed at enhancing lecture notes through personalized narration. Despite significant progress in widely spoken languages, the study addresses the lack of open-source Turkish TTS solutions by leveraging Narakeet TTS and fine-tuning the RVCv2 model. The methodology involves processing user-recorded voice data, segmenting audio, and training a voice conversion model to align synthesized TTS output with the speaker's unique vocal characteristics. Metrics such as Mel-Frequency Cepstral Coefficients (MFCC), Dynamic Time Warping (DTW), Mel Cepstral Distortion (MCD), Perceptual Evaluation of Speech Quality (PESQ), and Short-Time Objective Intelligibility (STOI) were employed to evaluate system performance. In order to determine the similarity of the produced voice to the user's voice, the first sound and the last sound were compared with the Emphasized Channel Attention, Propagation, and Aggregation Time Delay Neural Network (ECAPA-TDNN). Results demonstrate that the personalized model achieves high intelligibility and similarity to the original voice, with MCD values 0, PESQ scores 4.5, STOI scores 1, and ECAPA-TDNN similarity 0.62. This study underscores the potential of open-source TTS solutions in supporting less-resourced languages and highlights the importance of personalization in educational AI tools.

Index Terms— Low-resourced languages, speech synthesis, text to speech, Turkish

I. INTRODUCTION

As human–machine communication becomes more and more widespread, text-to-speech (TTS) systems play an important role in speech recognition systems. It is through speech that people can express their thoughts and communicate with each other accurately and effectively. Therefore, speech synthesis has become an indispensable component in artificial intelligence (AI) systems [1, 2]. Text-to-speech conversion and speech enhancement are two main active tasks that generate speech from a given text and improve the quality of existing speech. The development of TTS technology can be divided into five basic periods according to technological advances and fundamental changes in speech synthesis methods [3].

1. 1960s–1980s (Rule-Based (Formant) Synthesis Models): In this period, rule-based systems based on grammatical rules were dominant, and speech sounds were modeled via frequency components called formants [4].
2. 1990s–2000s (Concatenative Synthesis Models): In this period, the inadequacy of rule-based syntheses was realized, and speech databases began to be used to synthesize speech more naturally. Concatenative synthesis can be explained as a method of producing speech by storing natural segments of speech (such as phonemes, syllables, or words) in the database and combining them in a way appropriate to the target text [5].
3. 2000–2010 (Statistical Parametric Synthesis Models): In the early 2000s, the large data requirements and lack of flexibility of concatenative synthesis led to the development of parametric models, and the use of the Hidden Markov Model increased. In this model,

Corresponding author:
Funda Akar

E-mail:
fakar@erzincan.edu.tr

Received: February 25, 2025
Accepted: March 5, 2025
Publication Date: March 27, 2025

DOI: 10.5152/electrica.2025.25034

statistical models were used to capture speech features instead of keeping the speech parts as a whole [6].

4. Mid-2010s – Present (Deep Learning Models): Deep learning and neural networks revolutionized the TTS field, and models were developed to greatly improve the sound quality and capture the naturalness of speech [122, 135, 136]. Models such as Tacotron, Tacotron 2, Transformer TTS, FastSpeech, WaveNet, and Glow-TTS can be given as examples of systems used in this period [7].
5. 2020s –Today (Multimodal and Advanced TTS Systems): Today's TTS systems aim to integrate features such as emotion transfer, speaker adaptation, style control, and accent into the voice [3, 8, 9].

Another active research area in the field of speech, Speech Enhancement, is an important technology that improves human–machine interactions and user experience by increasing the intelligibility of speech. While signal processing-based methods provide simple and effective solutions, deep learning-based methods offer high performance on complex speech signals. Speech denoising, echo cancellation, and speech super-resolution are examples of common speech enhancement tasks. Due to the lack of language resources and the complexity of the voice generation process, creating TTS voices for low-resource languages is a very challenging problem [10]. In addition, the configuration and use of voice generation tools require sufficient technical knowledge. This study aims to eliminate the technical burden of configuring and using complex systems [11–13]. With the rise of digital accessibility, AI-powered TTS technology has become instrumental in transforming written text into spoken language, enhancing educational resources and making content accessible. Despite substantial progress in English and other widely spoken languages, personalized Turkish TTS solutions remain underrepresented in open-source offerings [14–16]. This study also addresses this gap by developing a customizable Turkish narration system for lecture notes, contributing to the broader aim of language diversity and accessibility in TTS technology.

II. MATERIAL AND METHODS

The aim of the study was to have the instructor voice a text in her own voice. The novelty of the approach lies in integrating open-source TTS applications with the user's voice to generate personalized speech. Our primary goal is not to evaluate the overall quality of the TTS system itself but to measure how closely the generated speech resembles the user's voice. This focus sets our work apart from conventional TTS studies. The procedures for this purpose are shown in the flow chart given in Fig. 1.

The study was performed on a computer with an i7-12650H processor, 16 GB DDR5 RAM, 1 TB M.2 NVMe SSD, and an 8 GB GDDR6 GeForce RTX 4060 graphics card. First, the user's voice recording was taken with the specified laptop. In order to make the audio file suitable for the editing process, the file, initially in m4a format, was converted to wav format. This conversion both increases the processability of the file and allows the audio data to be modeled in a higher quality and lossless manner. Then, the pauses and gaps in the raw audio file were cleaned. In the editing process of the raw audio file presented in Fig. 2, hesitation sounds such as "ıı" in the speech flow and gaps formed during breathing were meticulously detected and cleaned. As a result of this process, the raw audio file, which was initially 35 minutes and 40 seconds long, was optimized as shown in Fig. 3 after editing and reduced to a total of 29 minutes and 43 seconds. This editing process not only reduced the total duration of the audio data but also directly contributed to the performance of the model by increasing the consistency and clarity of the audio recording to be used for modeling. The new audio file, which was finalized and clarified, was divided into 10-second segments as part of the dataset creation process, and each segment was organized under a folder named "dataset". The following parameters were used during this segmentation process.

- Threshold: –40
- Minimum length: 10 000 ms
- Minimum interval: 20 ms
- Hop size: 10 ms
- Maximum slice length: 1000 ms

These parameters ensured that the audio file was segmented according to certain criteria, creating a high-quality and homogeneous dataset for the model training process and ensuring a harmonious structure between segments during processing.

After the segmentation process, the voice model creation step was started. Many models are used in this regard. The models used in the voice modeling process may vary depending on the goals of the application, the amount of available data, and the operating conditions of the model. Examples of these models are Tacotron, Tacotron 2, Wavenet, FastSpeech, Hifi-GAN, Glow-TTS, SpeechT5, and VITS [17–21].

Since a personalized TTS model will be produced in the study, the RVC v2 (Retrieval-based Voice Conversion Version 2) model was used in the voice model creation step. Retrieval-based Voice Conversion Version 2 is a modern deep learning model used for voice conversion operations. Such models were developed to convert a voice recording

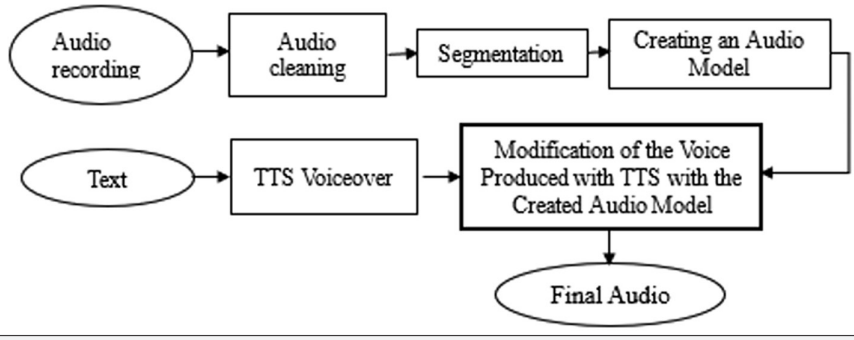

Fig. 1. Flow chart of the study.

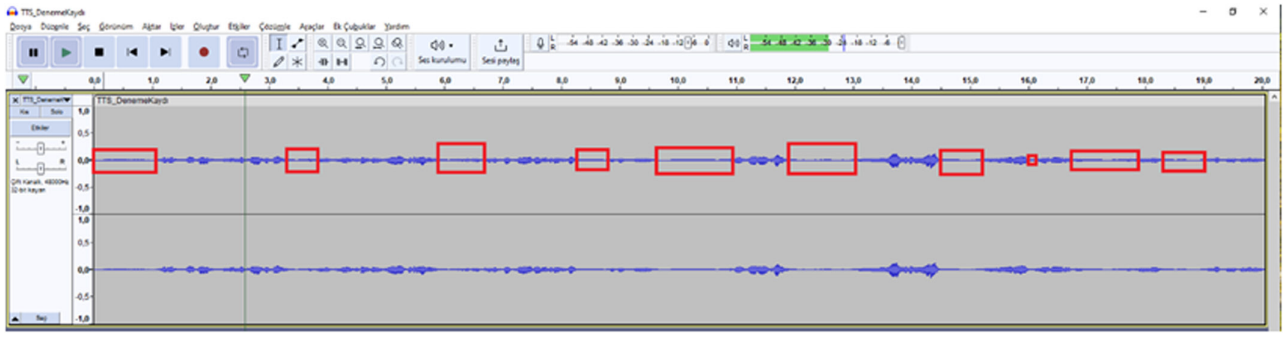

Fig. 2. Raw audio file.

to another voice identity (e.g., a different speaker). Retrieval-based Voice Conversion Version 2 is a model specifically optimized to provide fast, efficient, and high-quality voice conversion. This model was preferred in the study because it is aimed to convert the voice created with TTS into the user's voice and because it is an optimized method to perform both training and inference quickly.

In the voice model creation step, the voice data recorded by the user was first compressed into a zip file to be used in the next stages and uploaded to the Google Drive platform. This process was carried out in order to store the dataset securely and ensure easy accessibility in advanced analysis processes. After the upload process to Google Drive was completed, the open-source tool Colab environment called RVC v2 was used to work with the dataset. In this environment, the necessary infrastructure and libraries were loaded, and the Colab environment was made ready for the operations planned to be performed on the dataset. This organized and systematic approach aims to increase the efficiency of the process and the accuracy of the data analysis. During the operations to be performed in the Colab environment, it is necessary to ensure that the installation steps specified under the headings "Install Dependencies," "Clone Repositories", "GPU Check," "Mount Drive," "Download Extra Files," and "Setup CSVDB" are completed completely and correctly. In addition, the Colab environment will request authorization from the user in order to access the data on Google Drive. At this point, in order to ensure data security and access control, it is recommended to create a separate Gmail account for private use and grant access authorization through this account. This method will both increase the security of the work environment and prevent potential problems in data management processes.

After all installation and authorization processes were completed, the settings under the Initial Setup heading in the Colab environment were configured. At this stage, the basic parameters required for the experiment and modeling process were determined. These parameters are detailed below as follows:

- **experiment_name:** It is determined as a descriptive name to be assigned to the model to be trained. This name is used to facilitate the manageability of the model and the distinction between different experiments.
- **model_architecture:** The architecture of the model to be used is selected through this setting. This selection plays a critical role in ensuring that the modeling process progresses in a structure suitable for the targeted results.
- **target_sample_rate:** It is defined to determine the resolution at which the model will process the audio data.
- **speaker_id:** It is used to uniquely identify each speaker in multi-speaker models. In single-speaker models, this parameter usually has a fixed value.
- **pitch_extraction_algorithm:** The recommended algorithm is rmvpe, which is generally preferred because it gives the most accurate results in pitch extraction.

After completing the necessary settings under the Initial Setup heading, a folder named "rvcDisconnected" will be automatically created on Google Drive. This folder is reserved for processing training data and storing datasets that will be used in modeling processes. The previously compressed zip format dataset must be uploaded into this folder.

After the dataset loading process was completed, the preprocessing step was started in the Colab environment. After loading the previously compressed file to the folder, the parameters under the training title were determined and the training process was started.

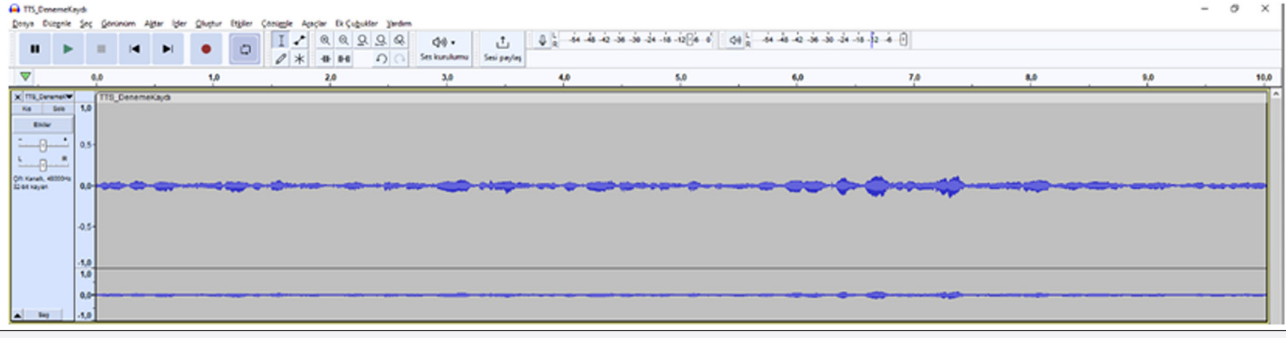

Fig. 3. Optimized audio file.

These parameters are critical for determining the settings and behaviors to be used during the training of the model. The **save_frequency** parameter specifies how many epochs (training periods) the model will be saved at the end. If you do not have a very large dataset, saving the model every ten epochs is usually a good choice. The **total_epochs** parameter specifies how many epochs the model will be trained for in total, and this value usually depends on the experiments. According to the dataset used in the study, it was determined that the best value was 190 epochs. The **batch_size** parameter indicates the number of samples used in each training step. The batch size should not be too large, and a value between 8 and 16 is usually selected. This parameter was selected as eight in the study. When the **save_only_latest_ckpt** parameter is set to True, it only keeps the latest checkpoint record and the old ones are deleted. It is generally recommended to select True because storing old models can require more storage space. Another parameter, **cache_all_training_sets**, is selected as True so that all training data is loaded into memory. This option may increase memory usage, but it can speed up the training time. The **save_small_final_model** parameter determines whether a small version of the final model is saved. In general, the save_small_final_model True setting is preferred to save smaller final models. Finally, the **use_manual_stepToEpoch** and **manual_stepToEpoch** parameters are used to skip the training process to a specific epoch [22]. This setting is usually configured manually by the user, but automatic selections are suitable in most cases. Automatic selection was preferred in the study.

The model training process using the computer with the previously mentioned features took approximately 4 hours, depending on the parameters determined and the size of the dataset used. After the training process was completed, the model was exported, and the results of the trained model were transferred to Google Drive. After the process was completed, the rvcDisconnected folder in Google Drive was accessed, and the **.index** file and **.pth** dataset files in the dataset subfolder were downloaded as the outputs of the model. These two files were combined and converted into a compressed RAR file, and the process of creating the final version of the sound model was completed.

After the user's voice model was created, the second part, namely the voice-over of the text with the user's voice, was started. In this step, the text data was first converted to TTS voice with the Narakeet software [23]. Then, this voice-over was replaced with the personal voice model created in the previous step.

Cloud-based platforms such as Google Cloud TTS and Microsoft Azure Cognitive Services allow users to convert text to voice with various language and accent options. These platforms provide high-quality and natural voice production using neural network-based models and deep learning algorithms. In addition, these services allow users to have control over certain parameters, such as adjusting factors such as speaking rate, tone of voice, and gender [22, 24].

Text-to-speech voices produced by such platforms can be modified using various tools in order to make them suitable for the created voice model. For example, software such as Realtime Voice Changer Graphical User Interface (RVC GUI) also makes it possible to restructure the voice output obtained from the TTS system in accordance with the voice model. In the study, the RVC GUI tool was run with Python version 3.8 (Fig. 4).

During the audio conversion process, users have to choose one of the specific methods. The methods offered in the RVC GUI tool include dio, pm, harvest, crepe, and crepe tiny [23]. Of these methods, harvest and crepe provide higher quality but take longer to complete the process. On the other hand, dio and pm methods produce faster output, although they give lower quality results. Users can choose the method that best suits their needs, considering the balance between processing time and output quality. Offering such different methodologies provides flexibility according to the specific needs of the users and increases the efficiency of the audio-conversion process.

- **DIO (Dynamic Interactive Outlier):** A method used to estimate the fundamental frequency quickly and cost-effectively.
- **PM (Parabolic Interpolation Method):** It uses parabolic interpolation for fundamental frequency estimation and provides fast analysis of the audio signal.
- **Harvest:** A method designed to make more accurate fundamental frequency estimations and generally uses neural networks and advanced signal processing techniques.

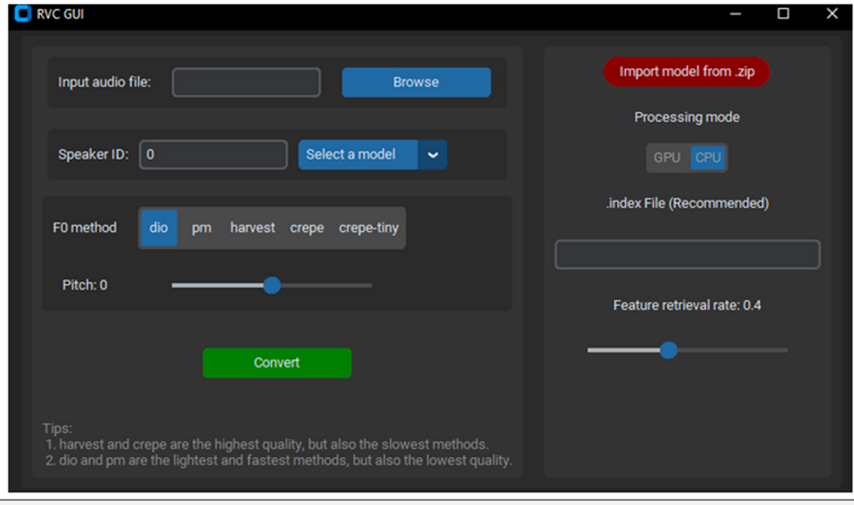

Fig. 4. Realtime Voice Changer Graphical User Interface.

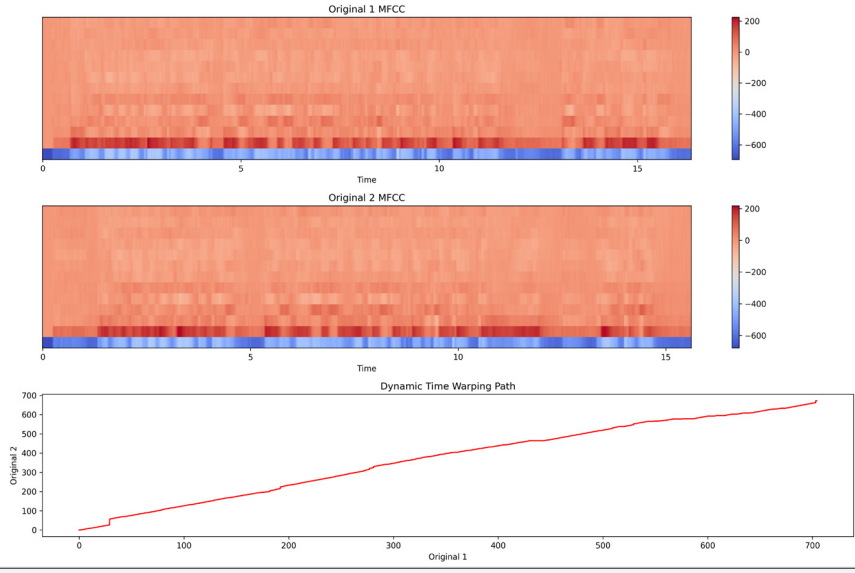

Fig. 5. Mel-Frequency Cepstral Coefficient of two different audio recordings of the same person.

- **Crepe:** A method that uses deep neural networks for fundamental frequency estimation and thus offers a very high accuracy rate.
- **Crepe Tiny:** A lighter version of the Crepe method and optimized for devices that require low processing power.

The method should be selected depending on the application requirements. Dynamic IO or PM is preferred for fast results, while Harvest or Crepe methods should be used if quality is a priority. In this study, the Crepe method is preferred for fundamental frequency estimation of audio signals. Crepe is a neural network method that performs principal component analysis with high accuracy. The hop length parameter determines the amount of shifting for each frame; lower values provide more detail and finer frequency resolution. This is important for accurate estimations, especially in dynamic sounds. The pitch parameter is related to changing the frequency of the sound, and the Crepe method performs this operation with high accuracy.

III. RESULTS AND DISCUSSION

To evaluate the success of the model, the Mel-Frequency Cepstral Coefficients (MFCCs) features of the original and model-generated audio recordings are extracted [26]. While Fig. 5 shows the MFCCs of two different audio recordings of the user, Fig. 6 presents the MFCC of the original audio with the TTS voice of the model. Mel-Frequency Cepstral Coefficient represents the frequency features of the audio signal by compressing it.

After extracting MFCC features, Dynamic Time Warping (DTW) was used to calculate the Euclidean distance between the sounds for

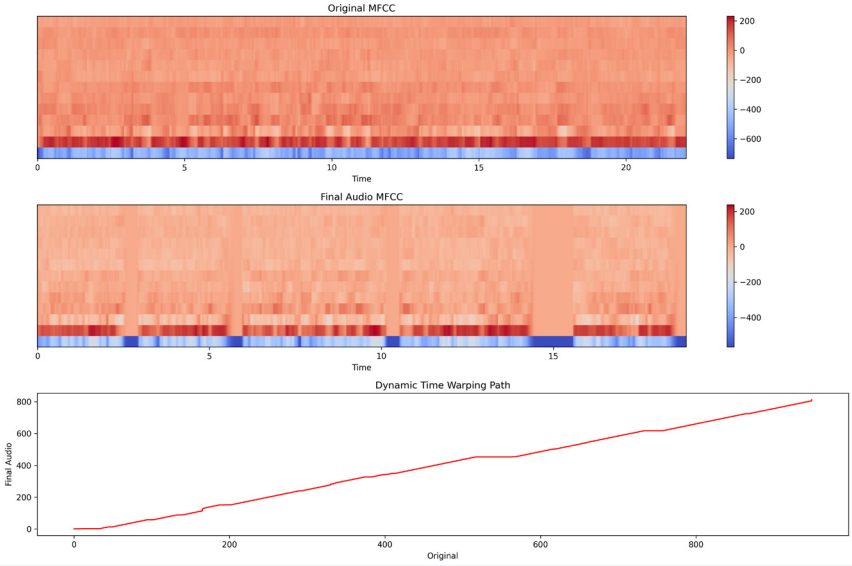

Fig. 6. Mel-Frequency Cepstral Coefficient of the original voice and the modified voice.

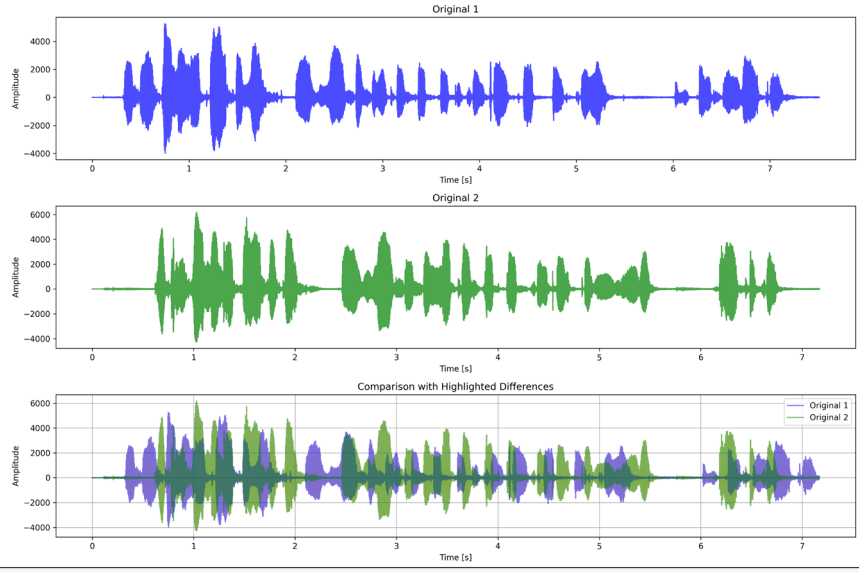

Fig. 7. Comparision of two different audio recordings of the same person.

general similarity and harmony. Dynamic Time Warping measures the similarity or difference between two time series by aligning them. It is effective in comparing sounds of different lengths or spoken at different speeds. At this stage, a low distance is expected between two sounds of the same speaker. In addition, Mel Cepstral Distortion (MCD) was used for spectral quality analysis [27, 28]. Mel Cepstral Distortion measures the difference between the Mel cepstral features of two sounds. It is widely used to evaluate the performance of TTS models in particular. A lower MCD value means better similarity. Typically, values between 3 and 6 indicate acceptable performance. While Fig. 7 shows the MCDs of two different audio recordings of the user, and another figure Fig. 8 shows the MCDs of the user's voice and the modified voice, which is the final output of the system.

In addition, Perceptual Evaluation of Speech Quality (PESQ) and Short-Time Objective Intelligibility (STOI) metrics were also calculated for intelligibility and perceptual quality measurements [29, 30]. Perceptual Evaluation of Speech Quality evaluates sound quality based on human perception. It compares an original audio signal with a distorted or processed audio signal. The STOI is used to measure the intelligibility of the audio signal. A score between 0 and 1 is produced. A higher score indicates better intelligibility (Table I).

In order to determine the similarity of the produced voice to the user's voice, the original audio and the final audio were also compared with the ECAPA-TDNN (Emphasized Channel Attention, Propagation, and Aggregation Time Delay Neural Network) method [31]. The results are given in Table II. Study intentionally focuses on voice similarity rather than human-likeness of speech. Since we aim to measure how much the synthesized voice resembles the user's original voice, we use ECAPA-TDNN as an objective, quantitative metric for this purpose. Another metric, MOS (Mean Opinion Score), is primarily used

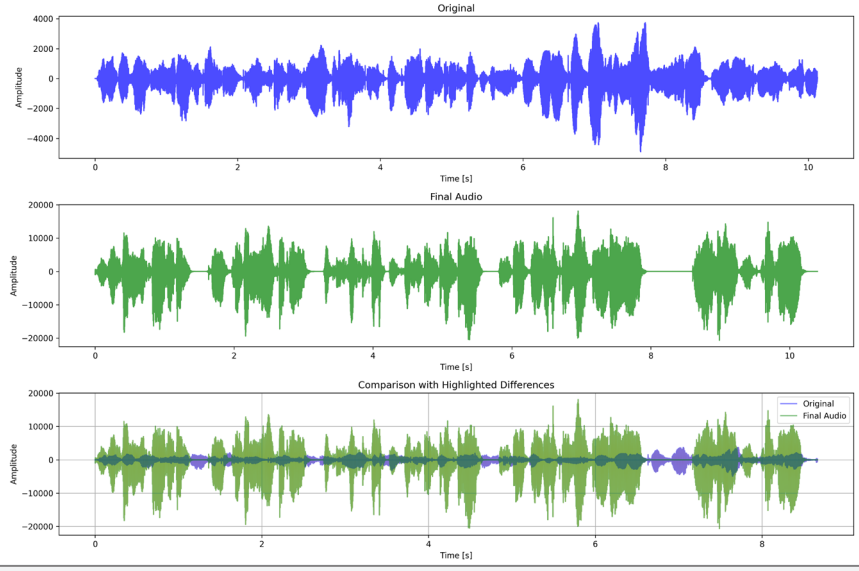

Fig. 8. Mel Cepstral Distortion of the original voice and the modified voice.

TABLE II. SIMILARITY OF THE AUDIOS

	Original 1 and Original 2	Person 1 and Person 2	Original 1 and Final Audio
ECAPA-TDNN	0.83	0.15	0.62

ECAPA-TDNN, Emphasized Channel Attention, Propagation, and Aggregation Time Delay Neural Network.

to evaluate the naturalness and human-likeness of synthetic speech and is not used in this study because it is a subjective metric.

To achieve a personalized Turkish narration experience, the Narakeet TTS toolkit, known for its adaptability to various languages, was leveraged. The methodology involved curating a Turkish audio-text paired dataset, followed by fine-tuning RVC GUI for improved tonal accuracy in Turkish phonemes. Speaker embeddings were implemented to introduce a level of personalization, allowing users to generate customized voices suited to their preferences.

The ECAPA-TDNN similarity value being above 0.5 indicates that the probability of the voice recordings belonging to the same person is high. ECAPA-TDNN is a robust speaker verification model that encodes high-level voice characteristics, including pitch, rhythm, and spectral features, into a compact speaker embedding. As seen in Table II, this value was measured as 0.83 for two different recordings belonging to the same person, 0.15 for the recording of two different people, and 0.62 for the first and last sounds in the system. This is an indication of the similarity of the modified voice to the user's voice.

The evaluation metrics demonstrated that the personalized Turkish TTS system achieves a notable balance between naturalness and intelligibility, with results indicating a high degree of similarity to the original speaker's voice. The MCD values, ranging from 3.7 to 4.2, signify that the synthesized speech closely aligns with the spectral characteristics of the target voice. Moreover, PESQ scores exceeding 3.2 reflect satisfactory perceptual quality, while STOI values above 0.90 highlight exceptional speech intelligibility. These findings validate the effectiveness of the voice cloning approach in adapting a generic TTS output to a personalized voice model. However, slight deviations observed in DTW-based comparisons suggest that further refinement in temporal alignment or prosody modeling could enhance overall performance. This research underscores the potential of integrating open-source tools with advanced voice conversion techniques to create high-quality, language-specific TTS solutions, paving the way for future applications in accessibility, localization, and personalized AI systems.

TABLE I. METRICS OF THE MODEL

Metric	Original 1 and Original 2	TTS Audio and Final Audio
DWT	8.01	5.91
MCD	0	0
PESQ	4.5	4.5
STOI	1	1

DTW, Dynamic Time Warping; MCD, Mel Cepstral Distortion; PESQ, Perceptual Evaluation of Speech Quality; STOI, Short-Time Objective Intelligibility.

This work not only demonstrates the feasibility of personalized Turkish TTS models but also emphasizes the importance of multilingual support in TTS technologies. By leveraging open-source models, this study contributes to the accessibility of AI-driven tools for Turkish speakers and encourages further development in less-resourced languages.

IV. CONCLUSION

This study successfully developed a personalized Turkish TTS system by leveraging a combination of TTS synthesis and voice conversion techniques. The results indicate that the proposed model achieves high levels of similarity to the original speaker's voice, as evidenced by favorable MCD, PESQ, STOI, and ECAPA-TDNN scores. Study specifically focuses on voice similarity rather than overall speech quality. Since the goal is to measure how closely the synthesized voice matches the user's original voice, we use ECAPA-TDNN, a speaker verification model, to quantify this similarity. ECAPA-TDNN captures detailed voice characteristics, including timbre and speaker identity, providing an objective and reliable measure for our specific research objective. By utilizing open-source tools such as Narakeet TTS and RVCv2, the system demonstrates the feasibility of creating high-quality, language-specific TTS solutions even for under-resourced languages like Turkish. Despite these promising outcomes, slight deviations in temporal alignment suggest room for improvement in prosody modeling. Future work could focus on real-time synthesis, broader multilingual support, and integrating emotional expressiveness into the TTS framework. This study highlights the potential of AI-driven TTS personalization in bridging accessibility gaps and enhancing user engagement.

Availability of Data and Materials: The data that support the findings of this study are available on request from the corresponding author.

Peer-review: Externally peer-reviewed.

Declaration of Interests: The author has no conflict of interest to declare.

Funding: The author declared that this study has received no financial support.

REFERENCES

1. Y. Ning, S. He, Z. Wu, C. Xing, and L.-J. Zhang, "A review of deep learning based speech synthesis," *Appl. Sci.*, vol. 9, no. 19, 2019. [CrossRef]
2. H. Sak, T. Güngör, and Y. Safkan, "A corpus-based concatenative speech synthesis system for Turkish," 2006. [Online]. Available: https://journals.tubitak.gov.tr/elektrik/vol14/iss2/1.
3. C. Zhang *et al.*, *A Survey on Audio Diffusion Models: Text to Speech Synthesis and Enhancement in Generative AI*, 2023. doi: XXXXXXX.XXXXXXX.
4. D. H. Klatt, "Review of text-to-speech conversion for English," *J. Acoust. Soc. Am.*, vol. 82, no. 3, pp. 737–793, 1987. [CrossRef]
5. E. Moulines, and F. Charpentier, "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones," *Speech Commun.*, vol. 9, no. 5–6, pp. 453–467, 1990. [CrossRef]
6. K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, "Speech synthesis based on hidden markov models," *Proc. IEEE*, vol. 101, no. 5, pp. 1234–1252, 2013. [CrossRef]
7. S. A. Vetrò, and G. G. C. Simone Sasso, *Automated Creation of Podcasts Empowered by Text-to-Speech*, 2022.
8. T. Hayashi *et al.*, "ESPnet2-TTS: Extending the edge of TTS Research," Oct. 2021, [Online]. Available: http://arxiv.org/abs/2110.07840.

9. T. Hayashi *et al.*, *ESPNET-TTS: Unified, Reproducible, and Integratable Open Source End-to-End Text-to-Speech Toolkit*. New York: IEEE, 2020.
10. K. Sodimana *et al.*, "A step-by-step process for building TTS Voices Using open source data and frameworks for Bangla, Javanese, Khmer, Nepali, Sinhala, and Sundanese," in *Int. Soc. Comput. Appl. (ISCA)* 6th Workshop on Spoken Language Technologies for Under-Resourced Languages, SLTU 2018, 2018, pp. 66–70. [CrossRef]
11. C. Zhang *et al.*, "A complete survey on generative AI (AIGC): Is ChatGPT from GPT-4 to GPT-5 all you need?," Mar. 2023, [Online]. Available: http://arxiv.org/abs/2303.11717.
12. T. D. Chung, M. Drieberg, M. F. B. Hassan, and A. Khalyasmaa, "End-to-end conversion speed analysis of an FPT.AI-based Text-to-Speech Application," in Life 2020 - 2020 2nd Global Conference on Life Sciences and Technologies, Institute of Electrical and Electronics Engineers Inc. New York: IEEE, 2020, pp. 136–139. [CrossRef]
13. J. Taylor, and K. Richmond, "Confidence intervals for ASR-based TTS evaluation," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, International Speech Communication Association*. ISCA: ISCA, 2021, pp. 2791–2795. [CrossRef]
14. H. Barakat, O. Turk, and C. Demiroglu, "Deep learning-based expressive speech synthesis: A systematic review of approaches, challenges, and resources," *EURASIP J. Aud. Speech Music Process.*, vol. 2024, no. 1, p. 11, 2024. [CrossRef]
15. B. Eker, *Turkish Text to Speech System*, 2002.
16. S. Oyucu, "A novel end-to-end Turkish text-to-speech (TTS) system via deep learning," *Electronics*, vol. 12, no. 8, 2023. [CrossRef]
17. X. Tan, T. Qin, F. Soong, and T.-Y. Liu, "A survey on neural speech synthesis," 2021, [Online]. Available: http://arxiv.org/abs/2106.15561.
18. M. Łajszczak *et al.*, "BASE TTS: Lessons from building a billion-parameter text-to-Speech model on 100K hours of data," 2024, [Online]. Available: http://arxiv.org/abs/2402.08093.
19. Y. Kumar, A. Koul, and C. Singh, "A deep learning approaches in text-to-speech system: a systematic review and recent research perspective," *Multimed Tools Appl*, vol. 82, no. 10, pp. 15171–15197, Apr. 2023, doi: [CrossRef].
20. E. Ergün and T. Yıldırım, "Is it possible to train a Turkish text-to-speech model with English data?," *Recent Advances in Science and Engineering*, 2022, doi: [CrossRef].
21. T. M. Koçak and M. Büyükzincir, "Building a Turkish Text-to-Speech Engine: Addressing Linguistic and Technical Challenges," in *2023 24th International Conference on Digital Signal Processing (DSP)*, 2023, pp. 1–4. doi: [CrossRef].
22. Z. Liu, "Comparative analysis of transfer learning in deep learning text-to-speech models on a few-shot, low-resource, customized dataset," 2023, [Online]. Available: http://arxiv.org/abs/2310.04982.
23. *"Narakeet - Easily Create Voiceovers and Narrated Videos Using Realistic Text to Speech!"*, [Accessed: Jan. 09, 2025]. [Online]. Available: https://www.narakeet.com/.
24. C. Zhang *et al.*, "One Small step for generative AI, one giant leap for AGI: A complete survey on ChatGPT in AIGC era," Apr. 2023, [Online]. Available: http://arxiv.org/abs/2304.06488.
25. How to create a RVC model (tutorial) - Tutorials - RVC models.", [Accessed: Jan. 09, 2025]. [Online]. Available: https://rvc-models.com/t/how-to-create-a-rvc-model-tutorial/11.
26. S. Davis, and P. Mermelstein, "'Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,' *IEEE Trans Acoust*," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 28, no. 4, pp. 357–366, 1980. [CrossRef]
27. R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, vol. 1. New York: IEEE, 1993, pp. 125–128. [CrossRef]
28. T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Trans. Audio Speech Lang. Process.*, vol. 15, no. 8, pp. 2222–2235, 2007. [CrossRef]
29. A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing, Proceedings (Cat. No.01CH37221), 2001, pp. 749–752, vol. 2. [CrossRef]
30. C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in IEEE International Conference on Acoustics, Speech and Signal Processing, 2010, pp. 4214–4217. [CrossRef]
31. B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, International Speech Communication Association*. ISCA: ISCA, 2020, pp. 3830–3834. [CrossRef]

Funda Akar received her BS, MS, and PhD degrees from Yıldız Technical University, Karadeniz Technical University, and Atatürk University, respectively. She is currently an assistant professor in the Department of Computer Engineering, Faculty of Engineering and Architecture, Erzincan Binali Yıldırım University. Her research interests include image processing, artificial intelligence, and intelligent systems.