Available online at www.garmian.edu.krd

Passer Journal of Basic and Applied Sciences

Passer 8 (Issue 2) (2026) 712-722

Original Article

Deep Learning-Based Text-to-Speech Synthesis for Central Kurdish: A Transformer-Enhanced Approach

Salman Burhan Ahmed[1], Sozan Abdullah Mahmood[*1]

[1]*Computer Department, College of Science, University of Sulaimani, Sulaymaniyah, Iraq.*

Received 20 November 2025; revised 7 February 2026;
accepted 27 February 2026; available online 03 July 2026

DOI: 10.24271/PSR.2026.561069.2440

ABSTRACT

Speech synthesis technology has become widely used for education, business, and human-computer interaction. However, the development of a Kurdish language Text-to-Speech (TTS) system has not kept pace with that of other languages, particularly English. Traditional approaches to TTS, such as concatenation and statistical parametric, each have several problems, including unnatural sounding speech, limited flexibility, and a lack of emotion. This study presents an advanced autoregressive sequence-to-sequence neural TTS model for Central Kurdish, adopting a convolutional neural network (CNN)-Transformer encoder. The new encoder offers multi-head self-attention mechanisms with positional encodings for better handling longer dependencies and richer prosody in Kurdish speech. This hybrid architecture combines the powerful autoregressive decoder derived from Tacotron 2 with efficient sequence modeling by transformers while keeping the attention-based alignment mechanism. A Kurdish speech dataset was prepared by segmenting longer utterances to enhance the quality of synthesis and training stability. The proposed model, with sharper attention alignments and more precise prosodic modeling, outperforms the baseline. It emerged as the top-performing system, achieving a Mean Opinion Score (MOS) of 4.74 and a substantial reduction of Word Error Rate (WER) of 3.1%. This result establishes a new benchmark for Kurdish TTS and demonstrates that selective architecture enhancement can yield substantial benefits. The study highlights that the deep learning-based enhancement of Kurdish TTS systems is a means to close an important gap in this language technology, as such findings can be applied to other low-resource languages with high morphological richness.

Keywords: TTS, Central Kurdish, deep learning, end-to-end autoregressive TTS, Transformer encoder, Tacotron 2, WaveGlow.

1 Introduction

Speech synthesis converts written text into audible speech using computational models. Despite decades of development, TTS remains vital for interaction between humans and computers. Screen readers and voice assistants for users who are visually impaired, automated call platforms, and the procedures involved in the generation of multimedia content are all largely dependent on this technology.

Initially, TTS systems were built on rule-based frameworks that allowed for the conversion of written text into spoken language by utilizing predetermined phonetic rules and organized linguistic algorithms [1,2]. As research in the field progressed, concatenative synthesis emerged as a widely adopted technique. It generated speech by combining small units such as phonemes, syllables, or words. Concatenative synthesis produced more natural speech but required complex processing and extensive data. The synthesized speech lacked prosody and clarity, resulting in a robotic and unnatural sound.

While the rapid advances in text-to-speech technologies have led to many commercial applications, building a high-quality Kurdish language TTS system continues to pose challenges. For example, while earlier methods such as concatenative synthesis [1] and Statistical Parametric Speech Synthesis (SPSS) [2] have achieved success in creating text-to-speech systems for numerous other languages, they also produced output that missed the naturalness of spoken language and required substantial linguistic resources to create voice quality. When creating actual synthesized speech using DNN technology [3], while it produced improvements in overall synthesis quality, it is still necessary to have large datasets and enough computation capability to create high-quality synthesized voice output.

* Corresponding author
E-mail address sozan.mahmood@univsul.edu.iq (Instructor).
Peer-reviewed under the responsibility of the University of Garmian.

The contemporary end-to-end TTS architectures, such as Tacotron 2 [4] and FastSpeech 2 [5], have the ability to create natural-sounding speech without requiring any specific linguistic rules or complex feature engineering. The training of these systems requires a great deal of data that is well-curated and includes a wide variety of labelled speech. Current Kurdish TTS projects, either statistical-based [2] or rule-driven, are facing many challenges, including limited ability to model prosody and produce imprecise phonetic output. Tacotron 2 and Deep Voice 3 [6] are autoregressive methods that offer excellent quality audio output, but are relatively slow in terms of inference time. In recent studies, the exploration of non-autoregressive architectures, as well as GAN (Generative Adversarial Network)-based neural vocoders such as HiFi-GAN [7], has increased in an effort to achieve faster speech synthesis while maintaining high audio quality through parallel generation rather than sequential processing.

The research demonstrates a TTS system for Central Kurdish specifically Sorani dialect, addressing the ongoing need for high-quality synthesis of the languages with few resources. Recent studies have also explored machine learning and deep learning methods for Kurdish language processing, showing the potential of data-driven approaches for improving computational models in low-resource settings [8,9]. To improve long-range dependency modeling and prosodic expressiveness, a Transformer-based encoder [10] is integrated into the Tacotron 2 architecture, replacing the original recurrent layers. The convolutional preprocessing layers are retained, enabling the model to capture local acoustic patterns, while the Transformer enhances global contextual understanding. The rest of the architecture including attention, decoder, and vocoder follows the original Tacotron 2 design. The system is evaluated using different metrics, manifesting the increased naturalness of the TTS system that is deep learning-based compared to the traditional one.

This study aims to contribute to the improvement of TTS systems for underrepresented languages by attaining these goals and developing a strong model for future TTS development for additional Kurdish dialects. The primary contributions of this research might be summarized as follows:

- Development of a new TTS system for Kurdish based on Transformer-based self-attention encoders, combining Tacotron 2 decoder and attention-based modules to produce an overall TTS system.
- Training on a Kurdish data set containing 13.63 hours of voice recordings in total, whose samples greater than 12 seconds were segmented to enhance the stability of the training.
- Comparative study of Transformer encoder vs. CNN-BiLSTM baseline highlighting better alignment and modeling of prosody.
- Suggesting that the TTS system proposed here can be used efficiently in the field of education, communication aids, and computer media, thus aiding and promoting technology for the Kurdish language in several domains.

2 Literature Review

This section presents the evolution of TTS technology, from early rule-based and concatenative frameworks to cutting-edge deep learning systems. The study of the Kurdish Language is emphasized in the research, and the review contains an overview of each of the primary models developed, their efficacy, and how recent developments have improved the naturalness, clarity, and superior quality of the synthetic voice produced through Natural Language Processing (NLP).

2.1 Traditional TTS Systems

The methods of synthesis most widely used in traditional TTS systems were primarily built around a few basic approaches, which attempted to create synthesized speech by using specific acoustic or linguistic rules for their approximation to that of a human voice. Formant synthesis used a computer model of the resonant characteristics of the speaker's voice to generate speech, providing very fine control over each of the necessary acoustic parameters. However, results were generally robotic and unnatural in comparison to real-world human speech [11]. The approach of concatenative synthesis yields a more natural output by putting together small segments of pre-recorded speech-like phonemes, syllables, or diphones, selected from a large database [1]. This approach yielded clearer and more intelligible speech compared to rule-based approaches. Nevertheless, it required extensive data preparation and had frequent audible discontinuities at unit boundaries. Yet another classical method is articulatory synthesis, which reproduces speech by simulating the bodily movements of articulators such as the tongue and lips. However, the computational demands and complexity limited its practical use [12]. Although these early methods formed an important starting point for speech synthesis, they could not reproduce natural rhythm, prosody, and expressiveness of real human speech. These limitations, finally, gave rise to the use of modern data-driven models, which shall learn such patterns themselves.

In the context of the Kurdish language, numerous traditional TTS systems have been developed using unit-selection approaches. Various researchers have investigated allophone, syllable, and diphone-based concatenative models custom-designed for Kurdish phonetics [13]. The allophone-based system had the lowest perceived quality and was the most challenging to build because of its extensive unit inventory. Syllable-based systems offered better clarity and intelligibility, but it was the diphone-based models that consistently delivered the strongest overall performance.

A comparative study conducted in [14] found that the diphone-based approach generated the greatest quality, scoring 97% on the Diagnostic Rhyme Test (DRT). Further work with allophone

units within concatenative synthesis likewise reported a very high degree of intelligibility [15]. A later diphone-based system proposed in [16] further improved naturalness by minimizing spectral mismatches at unit transitions, yielding smoother and more realistic Central Kurdish speech.

These traditional Kurdish TTS approaches contributed much in terms of insights and developed an initial set of systems. However, their failure to accurately model prosody and expressive features, and dependence on large, manually curated databases, finally motivated the shift toward neural, end-to-end TTS frameworks.

2.2 Deep Learning TTS Systems

In recent years, deep learning has revolutionized text-to-speech synthesis, enabling models to produce speech that, in comparison to conventional methods, sounds far clearer and more natural. The first neural approaches replaced individual components of the TTS pipeline with neural networks, grapheme-to-phoneme conversion [17,18], pitch prediction [19], duration modeling [20], and waveform generation [21], which helped improve accuracy and reduce dependence on handcrafted linguistic rules. A major step forward came with WaveNet [22], a powerful autoregressive model capable of producing high-fidelity raw-audio waveforms. Although WaveNet achieved strong performance for TTS, its sample-level autoregressive nature made the inference slow and required conditioning on features from a separate TTS frontend, meaning it was not entirely end-to-end. When trained with 34.8 hours of Mandarin and 24.6 hours of English speech, WaveNet achieved MOS scores of 4.21 and 4.08, respectively.

Further improvements were introduced by DeepVoice [23], which replaced every stage of a traditional TTS pipeline with neural networks to improve speech quality. Shortly after, Tacotron [24] provided the first widely adopted end-to-end architecture for mapping text directly to mel-spectrograms using an attention-based sequence-to-sequence model. The results showed that Tacotron, which trained on 24.6 hours of North American English, achieved a MOS of 3.80. Char2Wav [25], another end-to-end model, used normalized WORLD vocoder features and was trained on the datasets VCTK and DIMEX-100, and produced clear and natural speech.

The work in Tacotron 2 [4] represented a significant milestone, where a Tacotron-style mel-spectrogram predictor was combined with a WaveNet vocoder. Trained with 24.6 hours of American English, this model achieved a MOS of 4.53, realizing very natural prosody and even clearer speech. WaveGlow [26] is a flow-based neural vocoder that enables fast and parallel waveform generation while maintaining high audio quality. Due to its training stability and inference efficiency, it has been widely adopted in end-to-end text-to-speech systems. Subsequent work presented a parallel wave generation technique called ClariNet [27], based on Gaussian inverse autoregressive flow (IAF), reaching a MOS of 4.15 with a much lower computational cost for vocoding.

Efficiency concerns led to the creation of FastSpeech [28], a non-autoregressive Transformer-based model. This brought FastSpeech to a MOS of 3.84 for the 24-hour LJSpeech, even though there are some restrictions on the audio quality, since it depends on teacher-student distillation, suffers from inaccurate duration prediction, and inherently loses some information during mel-spectrogram generation. Many of those issues were tackled directly by predicting pitch, duration, and energy in FastSpeech 2 [5]. Trained on LJSpeech, it reached a MOS of 3.83, offering more stable and faster synthesis.

Deep learning TTS has also been used successfully in the case of low-resource languages. For Myanmar TTS, Tacotron 2 was investigated in [29], where a training on 5 hours of speech yielded the MOS of 3.89. In the Arabic domain, a Tacotron 2 model, trained on 2.41 hours of the Nawar Halabi corpus on top of an English pretrained model, achieved a MOS of 4.21 [30]. For Persian, an enhanced Tacotron-based system trained on 21 hours of Persian language produced MOS scores of 3.01 to 3.97 across different test sets [31].

Deep learning-based methods have also started to be used in Kurdish TTS research. The work in [32] employs transfer learning from an English pre-trained Tacotron 2 model that was fine-tuned with 10 hours of Central Kurdish speech and achieved a MOS of 4.10. However, it was prone to issues caused by the alphabetic and phonetic inventory mismatches between English and Kurdish. More recently [33], presented a fully Transformer-based end-to-end Kurdish TTS system that employed adversarial training and a stochastic duration predictor. Its best achieved MOS was 3.94, while its computational requirement and model complexity were much higher.

Despite these advances, Kurdish TTS remains less explored than that of high-resource languages. Limitations in the dataset, inconsistent prosody modeling, errors in pronunciation accuracy, and a general need for simpler but effective architectures point to further improvement. This is a call for an effective model that can handle long-range dependencies, produce natural and high-quality speech, and remain practical for low-resource settings.

3 Data Source and Description

A TTS system requires a well-organized corpus, which should contain paired audio recordings and their corresponding text. Within this study, the Gigant-KTTS dataset [34] was used as the main training material for the proposed model and the WaveGlow vocoder. This has resulted in an overall natural-sounding Central Kurdish speech generated by the system. In summary, this dataset comprises 6,078 sentences, totaling 13.63 hours and covers 12 categories: sports, health, linguistics, news, and others. The original sentences included in this dataset had diverse lengths, and the duration of these sentences ranged from 0.502 to 16.781 seconds. Audio samples longer than 12 seconds were segmented carefully based on meaningful sentence boundaries to achieve the best training stability and performance of the model. Figure 1 presents the wide range of topics that were considered for the corpus after segmentation. Consequently, only segments from 0 to 12 seconds were retained, yielding a cleaned dataset of 7,297 sentences. The refined dataset was ultimately split into two parts,

with 6,567 utterances (90%) allocated for training and the remaining 730 utterances (10%) reserved for validation.

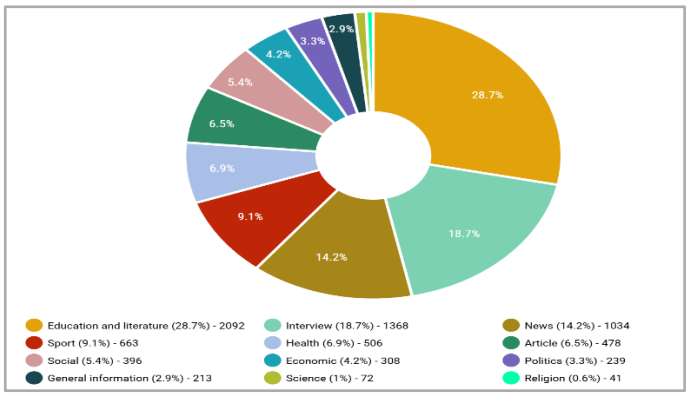

Figure 1: Overview of how the segmented Kurdish speech corpus is distributed across different topic categories.

The utterance duration of the processed dataset varies between 0.5 and 12 seconds, resulting in a wide range of audio lengths. The distribution of sentence length after expansion, as shown in Figure 2, is essential for effective TTS training, as it exposes the model to different speech tempos and intonation types. In addition, the shorter clips give the model a better opportunity to learn detailed acoustic characteristics, while the longer recordings provide the model with increased rhythmic structure and contextual fluidity (i.e., prosody, rhythm flow, continuity).

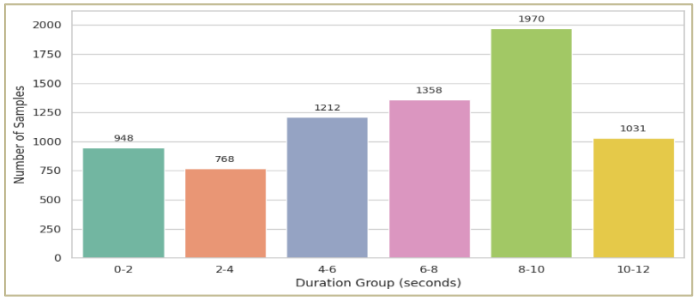

Figure 2: Duration distribution of audio samples in Gigant-KTTS after segmentation.

This dataset is very important in advancing TTS systems in being able to enable the generation of smoother and more human-like sounding utterances in a variety of environments and situations. Table 1 displays the properties of the segmented audio files.

Table 1: Summary of the key properties of the segmented Gigant-KTTS audio dataset.

No.	Attribute	Description
1	Recording Setting	Captured in a professional, low-noise studio environment
2	Speaker Profile	Male narrator with a clear and natural vocal style
3	Language	Central Kurdish (Sorani)
4	Total Audio Duration	Approximately 13.63 hours of speech
5	File Format	Stored in standard WAV format
6	Audio Channels	Mono output
7	Sample Rate	22.05 kHz sampling frequency
8	Bit Resolution	16-bit recordings
9	Minimum Clip Length	Shortest segment is 0.5 seconds
10	Maximum Clip Length	Longest segment is 12 seconds

4 Data Preprocessing

This section details the text and audio preprocessing steps required to ensure the Kurdish dataset is consistent and optimized for high-quality speech synthesis.

4.1 Text Preprocessing

Text data preparation is one of the most important components in building an effective TTS system. Although the Central Kurdish writing system is consistent, many spelling and stylistic issues exist due to significant variation among writers. Hence, text normalization aims to generate clean and standardized text inputs for TTS model training. Orthographic normalization was used to address spelling variations, e.g., harmonizing Arabic-based script and various Unicode characters in Kurdish text. Numerical values, such as dates and times, were spelled out to reproduce what people would intuitively read during synthesis. These preprocessing steps will therefore yield a cleaner, more consistent text corpus summarized in Table 2 that will help the model to generate clearer, more natural, and intelligible synthesized speech.

Table 2: Preprocessing operations applied to the Central Kurdish text.

Type of Preprocessing	Details	Example
Numeric and Date Conversion	Rendering numbers, dates, and time expressions into their fully spoken Kurdish forms.	"٢٠٢٥-١٢-٢٠" → "بیستی دوانزەی دووهەزاروبیست وپێنج"
Text Normalization	Normalizing different ways to spell the same word to ensure a uniform written	"كتيب" → "کتێب"

	form across the dataset.	
Punctuation Refinement	Cleaning excessive or misplaced punctuation and restructuring text for clear and coherent sentences.	"تکایە، وانەکان بەباشی. بخوێنە" → "تکایە وانەکان بەباشی بخوێنە"
Abbreviation Expansion	Converting shortened or abbreviated forms into their complete lexical equivalents.	"د." → "دکتۆر"

4.2 Audio Preprocessing

Effective text-to-speech synthesis requires not only high-quality, well-prepared text but also consistently processed audio data. Therefore, a number of preprocessing steps are applied to the raw speech recordings in this dataset to improve their quality and make them suitable for the TTS model. Table 3 provides a summary of these operations.

Table 3: Audio preprocessing operations applied to the speech corpus.

Processing Step	Summary
Silence Removal	Cuts unnecessary silence at the beginning or end of recordings.
Background Noise Reduction	Filters out hums, static, and other unwanted noise.
Unified Sample Rate	Converts all audio to a consistent 22,050 Hz sample rate. Typical TTS sampling rate of 22,050 Hz.
Volume Leveling	Normalizes volume so all clips have similar loudness levels.
Clip Length Management	Keeps audio segments within 0-12 seconds and splits longer ones.

4.3 Feature Extraction

Feature extraction is instrumental in enabling the TTS model to learn effective patterns from both audio and text. For the text input, a character-based approach is employed, in which the normalized sentences are decomposed into individual characters, and then mapped to a vocabulary of distinct IDs using a predefined set of characters. These sequences are then embedded in dense vector form and fed into the encoder. For audio, the raw waveform converted to mel-spectrograms, which represent time-frequency features of speech. The model was indeed trained on these spectrograms, where during training they were specified as output targets, allowing it to learn what acoustic structure to generate a natural-sounding synthesis.

5 Methodology

This section describes the methodology adopted in this study, including the proposed model architecture, training configuration, and evaluation procedures. The system is implemented as an end-to-end neural text-to-speech framework that integrates all components within a unified architecture.

5.1 Model Architecture

In this study, we introduce an improved sequence-to-sequence neural model incorporating a Transformer encoder with the attention and decoder components of the Tacotron 2 architecture to produce an end-to-end text-to-speech synthesizer. The proposed model converts text sequences into mel-spectrograms and generates corresponding speech waveforms using a neural vocoder, while the Transformer encoder enables more effective modeling of long-range dependencies and prosodic information in Kurdish speech. An overview of the proposed model structure is given in Figure 3. The improved model includes a sequence-to-sequence structure that generates mel-spectrograms from given input text and has three major parts: the Transformer-based encoder, attention, and an autoregressive decoder, combining natural prosody with efficient sequence modeling.

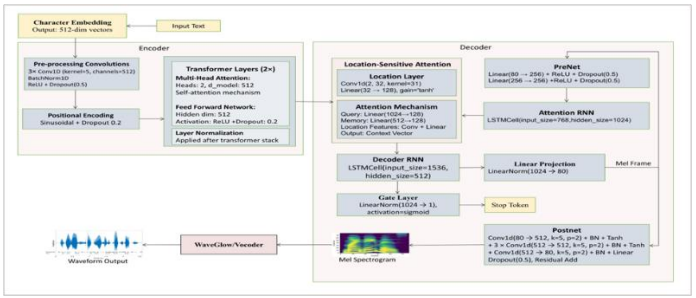

Figure 3: Proposed Model Architecture.

5.1.1 Transformer-based Encoder

The architecture feeds input text into a high-level pipeline, which starts from character embedding and ends with high-quality mel-spectrogram generation. Input text sequences are first transformed into learnable 512-dimensional character embeddings, initialized using a uniform distribution. These embeddings capture the linguistic characteristics of individual characters in the Kurdish alphabet.

Our primary architectural contribution replaces the baseline CNN-BiLSTM encoder with a CNN-Transformer design that better model's long-range dependencies. The encoder architecture incorporates a three-stage processing pipeline to simultaneously capture global and local feature representation. First, three consecutive one-dimensional convolutional layers with a kernel size of five are used to capture local contextual

patterns, each followed by batch normalization, ReLU activation, and dropout regularization. Following convolutional preprocessing, standard sinusoidal positional encoding is added to preserve sequence order information. These enriched features are passed to a pair of Transformer encoder layers. Each layer uses multihead self-attention with two heads, along with a feed-forward network that contains 512 hidden units. This model combines scaled dot-product attention, feed-forward networks activated by ReLU, residual connections, and layer normalization for stability during training, with dropout of 0.2 throughout and appropriate Xavier uniform weight initialization. Efficient modeling of long-range dependencies and nuanced prosodic structures, while providing the most efficient architectures (layer depth and attention heads), focused on the design of the proposed architecture, this approach provides an effective structure specifically tailored for Kurdish speech synthesis.

5.1.2 Attention Mechanism

A primary function of attention mechanisms is facilitating interaction between the encoder and decoder subcomponents. Additionally, location sensitive-attention enhances the performance of additive attention found in standard additive attention models through incorporating information related to both historical and current attention, thus enhancing the degree of alignment achievable, reducing common synthesis artefacts that arise when using additive attention alone. This superior attention framework resolves the simple issue of achieving correct text-to-speech alignment on long sequences without skipping and repeating artefacts, which are characteristic of autoregressive TTS models. Operating on a two-component architecture of content-based and location-based attention computation, the location layer calculates cumulative attention of past decoding steps by a 1D convolution operation with 32 filters of kernel width 31, and linear transforms resulting values into 128-dimensional location features. This convolution-based processing enables the model to maintain a spatial record of attention history, tracing which parts of the input sequence are covered and achieving a monotonic progress of the text during synthesis.

The attention computation aligns three major information sources to generate strong alignment energies. Decoder attention RNN-based query vectors (128-dimensional) hold the current decoding state, while processed Transformer encoder memory provide dense input sequences contextual representations. Such elements are combined with the location features by additive attention computation, where all three representations are projected into a 128-dimensional shared space and merged by element-wise addition followed by hyperbolic tangent activation. The computed energies are normalized using the softmax function in order to create attention weights for a particular distribution of attention on various sections of the input sequence. The use of multi-source attention calculation leads to a more accurate, stable and reliable text-to-speech alignment due to the use of both semantic content and temporal evolution to be considered when calculating such correlations. This increased accuracy is especially beneficial for the prosodic features necessary for Kurdish language speech synthesis.

5.1.3 Decoder

The decoder uses an autoregressive method for generating mel-spectrogram frames sequentially in a complex multi-stage system designed to maintain high temporal coherence and spectral quality. Generation begins with a pre-net made up of two layers of 256 units that are fully connected to each other (ReLU activation and 0.5 dropout), fed into previous mel-spectrogram frames, creating an information bottleneck that reduces over-reliance on past predictions, enabling robust attention learning.

The basic decoding mechanism consists of a two-stage LSTM network that operates on information in a hierarchical manner. First, the attention RNN is a single-layer 1024-unit LSTM that attends on the pre-net output and context vector, both merged, with 0.1 dropout on hidden states for its regularization. This component focuses on alignment computation and retention. Then there is the decoding RNN, a 512-unit LSTM to combine its output with the input context vector for computing the final hidden representation containing both temporal dependencies and contextual information for mel-spectrogram prediction.

Prediction is carried out through parallel linear transforms, which simultaneously generate 80-dimensional outputs and binary stop tokens (gates) signaling termination of sequences, with dynamic sequence length control. To maximize spectral quality, a convolution post-processing network refines these predictions using five 1D convolution blocks (512 channels, kernel 5) with batch normalization and dropout. This post-net generates residual corrections, added to the raw output, greatly adding detail to the resulting spectra, reducing artefacts, and overall optimizing the naturalness of synthesis. The predicted mel-spectrogram is then transformed into an audio waveform using a vocoder WaveGlow in this study, though alternatives such as WaveNet and HiFi-GAN are also viable.

5.1.4 Vocoder

WaveGlow acts as the vocoder in this system and is itself a flow-based generative model, which generates audio waveforms from mel-spectrograms in real-time. Based on the Glow architecture but modified for speech synthesis, it uses a sequence of 12 invertible flow steps to gradually shape simple Gaussian noise into natural-sounding speech conditioned on the input spectrogram. Each flow step includes components such as 1×1 invertible convolutions and affine coupling layers, thus allowing for efficient parallel computation rather than slow sample-by-sample generation. This model works at a 22.05 kHz sampling rate and generates 16-bit audio directly. In training, WaveGlow is optimized through a negative log-likelihood objective that enables it to discover the fundamental distribution of actual speech in a steady, end-to-end way. This design enables the vocoder to generate smooth and high-quality waveforms while keeping fast inference performance.

5.2 Training

This part explains how the speech synthesis model was trained, outlining the process used to generate mel-spectrograms with the acoustic model and reconstruct the corresponding audio using the WaveGlow vocoder. To assess the quality of our proposed Transformer-augmented encoder, we compared and trained two different architectural setups the baseline encoder versus our new Transformer-based encoder architecture.

5.2.1 Training the Acoustic Model

We trained both systems using end-to-end supervised learning to convert input text sequences into mel-spectrograms. The Kurdish dataset described in Section 3 was employed for training and evaluation. The refined dataset was split into training and validation sets as described in the data preparation stage.

The training process was performed on Google Colaboratory using a Tesla T4 GPU with 15 GB of VRAM. Both models are trained under the same hardware setup for comparison. Both models utilized the same training parameters to keep a level playing field. We utilized Xavier uniform initialization and trained with Adam optimizer with hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\varepsilon = 1e{-}9$, and weight decay of $1e{-}5$. Gradient clip was 1.0. Both models were trained for 240,000 steps using a batch size of 32, which corresponds to approximately 1,171 epochs through the training set, saving checkpoints every 300 steps and validating every 300 steps.

Teacher forcing is used during training, where ground-truth mel-spectrograms guided the decoder. The loss function, combining reconstruction errors and binary cross-entropy for stop-token prediction, is formally defined in Equation (1).

$$\boldsymbol{Total_{loss}} = \mathbf{MSE}\left(\hat{\boldsymbol{Y}}, \mathbf{Y}\right) + \mathbf{MSE}\left(\hat{\boldsymbol{Y}}^{\boldsymbol{postnet}}, \mathbf{Y}\right) + \boldsymbol{BCEWithLogits}\left(\hat{\boldsymbol{G}}, \mathbf{G}\right) \quad (1)$$

where $\hat{Y}$ denotes the output of the decoder, $\hat{Y}^{postnet}$ is the postnet-enhanced output, and $\hat{G}$ denotes the gate signal prediction. Y and G are the ground-truth mel-spectrogram and stop token, respectively. This loss formulation supports accurate spectral reconstruction and reliable sequence termination, avoiding over-generation or early stopping typical of autoregressive models. Figures 4 and 5 display the training and validation loss curves for both the baseline encoder and the one enhanced with Transformer. It can be seen that the loss values monotonically decrease during training. The convergence of the model using the Transformer is faster and more stable than that of the baseline.

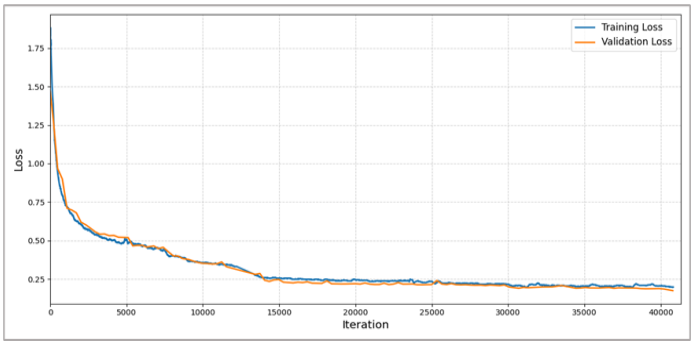

Figure 4: The proposed model training and validation loss plot.

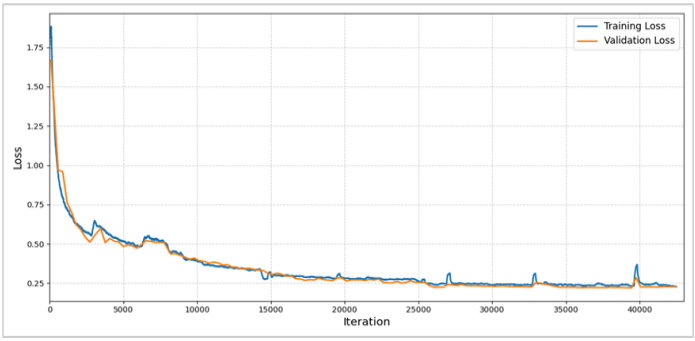

Figure 5: The baseline model training and validation loss plot.

We used positional encoding for order information in sequences, masking of variable-length sequences, and layer normalization to aid stability in training. Although its advantage, the Transformer architecture is found to be initialization and learning rate-scheduling, particularly during initial training periods. However, with appropriate tuning, it achieved more robust convergence and maintained alignment quality across longer sequences. We adopted a warm-up learning rate schedule for the Transformer model, given in Equation (2).

$$\boldsymbol{lr} = \boldsymbol{d_{model}^{-0.5}}.\boldsymbol{min}\left(\boldsymbol{step^{-0.5}}, \boldsymbol{step}.\boldsymbol{wramup\_step^{-1.5}}\right)$$

where d_{model}=512. This scheduling strategy enabled stable attention learning during the initial training phase. The baseline used a fixed learning rate consistent with the original setup.

The Transformer-based encoder demonstrated faster and stable convergence compared to the baseline. It reached a clear attention alignment pattern after around 15,000 steps, while about 25,000 steps that required by the baseline BiLSTM. This performance gain stems from the Transformer potential for representing global dependencies of input sequences without reliance on sequential recurrence. The proposed encoder had a number of advantages during training. Its self-attention had richer feature representations, leading to sharper and steadier attention alignment in the decoder. Visualizations always had neater diagonal patterns, indicating better text–audio alignment during training. Also, non-recurrent nature of the Transformer helped better gradient flow and training instabilities, which is particularly desirable in a morphologically dense language like

Kurdish. This architecture helped better modeling of longer-range dependencies and steadier loss reduction. Overall, the Transformer-augmented model outperformed the baseline by a significant margin in training stability, sharpness of attention, and convergence speed, and thus offers a strong candidate architecture of Kurdish TTS synthesis. Table 4 contains a list of all of the parameters of the proposed model.

Table 4: Training Hyperparameters for the proposed model.

Category	Parameter	Value
Encoder Parameters	Encoder kernel size	5
	Number of convolutions	3
	Embedding dimension	512
	Number of layers	2
	Feed-forward dimension	512
	Attention heads	2
	Dropout	0.2
Decoder Parameters	Frames per step	1
	RNN dimension	512
	Prenet dimension	256
	Max decoder steps	1200
	Gate threshold	0.5
	Attention dropout	0.2
	Decoder dropout	0.2
Attention Parameters	Attention RNN dimension	1024
	Attention dimension	128
	Location convolution filters	32
	Location kernel size	31
Mel-Post Processing Network	Postnet embedding dimension	512
	Number of convolutions	5
	Postnet kernel size	5
Optimization Hyperparameters	Weight decay	1e-5
	Gradient clipping threshold	1.0
	Batch size	32

An alignment graph offers an intuitive way to see how the model learns to connect written text with the sounds it produces. If the model is learning properly, the graph forms a clean diagonal line, showing that each character is being matched to the correct portion of the output speech. If the pattern appears scattered or unstable, it usually suggests that the model needs further training, or has not been able to determine adequate alignment yet. This diagonal structure only settles in when the model is past a number of training steps, as illustrated in Figures 6 and 7.

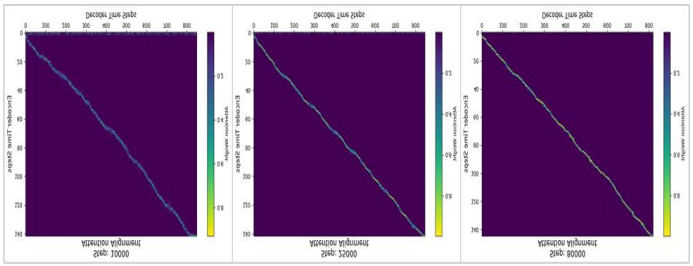

Figure 6: Proposed model attention alignment at certain steps.

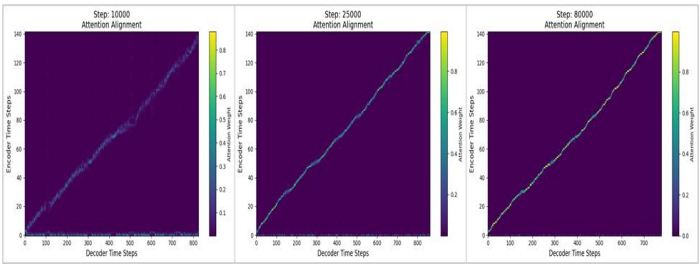

Figure 7: Baseline model attention alignment at certain steps.

5.2.2 Training the Vocoder

A WaveGlow vocoder was trained independently using the same dataset and 90/10 partitioning scheme described in Section 3 to ensure consistency in speaker and recording conditions. The vocoder followed a configuration of twelve flow blocks with a group size of eight. To accelerate convergence and stabilize training, the early outputs are extracted after every four flows, producing two channels at each early-output stage. The model incorporates affine coupling layers supported by a dilated convolutional stack of eight layers and 256 channels, drawing design principles from WaveNet. The same computational environment is used for vocoder training, with mixed precision computation of FP16, batch size 24, and learning rate 1×10^{-4}. It was trained for 230,000 steps, saving model checkpoints every 1,000 steps. Every phase of the training process resulted in the output from the vocoder to become exceedingly clear and true to life as shown by Figure 8 comparing raw audio data to the generated results from the model for several training steps and illustrating the stepwise improvement in the quality of the audio waveform over periods of time.

5.3 Evaluation

The subjective and objective methods were used in order to assess the performance of the proposed TTS system. In the subjective test, MOS test served as the primary perceptual metric, enabling native Kurdish listeners to judge the clarity and naturalness of each synthesized speech sample. For this purpose, scores between

1 and 5 were given (the best quality is 5). The average MOS was computed using Equation (3):

$$MOS = \frac{1}{N}\sum_{i=1}^{N} R_i \quad (3)$$

where R_i represents the individual rating for the i-th utterance, and N is the total number of evaluated samples. Statistical significance of MOS differences between systems was assessed using a paired samples t-test at the 95% confidence level ($\alpha = 0.05$). The paired design was adopted since each listener evaluated all systems on identical test samples, which controls for inter-rater variability and increases statistical reliability.

On the other hand, WER functioned as an objective assessment metric. As there is no reliable Kurdish ASR system, the generated audio was manually transcribed by native speakers. These transcriptions were compared with the original text to measure substitution, deletion, and insertion errors. WER is defined in Equation (4). Where S, D, and I stand for the different types of errors (substitutions, deletions, and insertions), and N is the total number of words in the original text.

$$WER = \frac{S + D + I}{N} \quad (4)$$

To test the system, more than 100 unseen sentences with their original recordings were prepared. A total of 50 native Sorani Kurdish speakers-28 males and 22 females, ranging in age from 20 to 50 years-old-participated in the evaluation. Most participants had university educations, and all used Kurdish as the main daily language. The group rated the baseline model, the proposed model, and the original human voice. The combination of MOS and WER results gave a clear picture of the system's quality from both human perception and measurable accuracy.

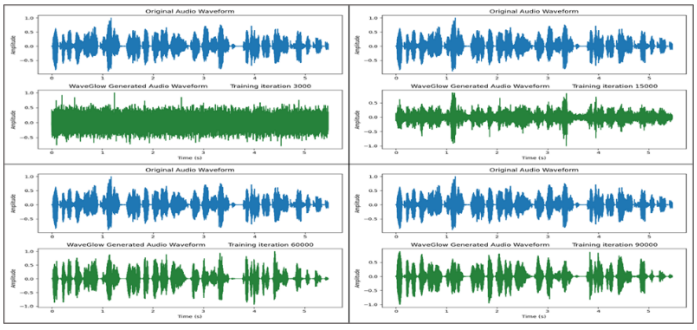

Figure 8: Comparison between the reference and the vocoder-generated waveforms at various training stages.

6 Results

Through a set of objective and subjective tests was conducted to evaluate the performance of the proposed Transformer-enhanced TTS model, with the outcomes compared directly with the baseline system. These evaluations were conducted to assess naturalness, intelligibility, and computational efficiency under the same conditions. These results strongly favor the new architecture. The proposed model achieved a MOS of 4.74 with a standard deviation (SD) of ±0.03, while the baseline scored 4.45 (±0.06), hence a gain of 0.29 in perceived naturalness by the proposed model. This improvement was found to be statistically significant ($p < 0.001$, paired t-test), confirming that the observed MOS difference is not due to random variation. This gain can be attributed to the Transformer encoder's ability to capture long-range contextual patterns and more expressive prosodic cues in Kurdish speech.

Objective intelligibility testing further reinforces these findings. In Table 5, it can be seen that the proposed system received a WER of 3.1% (±0.09), while the baseline model produced 7.8% (±0.11). This corresponds to a relative decrease in error of approximately 60%, indicating an improved realization of phonemes and better handling of Kurdish morphology.

Table 5: Speech quality evaluation (MOS and WER) for human reference, proposed and baseline models. Results are reported as mean ± standard deviation. Differences between the proposed and baseline systems are statistically significant ($p < 0.001$, paired t-test).

Metric	Human	Proposed Model	Baseline Model
MOS	4.98 (±0.02)	4.74 (± 0.03)	4.45 (± 0.06)
WER (%)	0.00	3.1% (± 0.09)	7.8% (± 0.11)

Apart from quality improvements, this model exhibited good computational efficiency. The proposed system generated an 8-second audio segment in 0.431 seconds, which corresponds to 0.0539 RTF during inference. The generation took 0.596 seconds with the baseline model, which gives an RTF of 0.0745. During inference, the memory usage for both systems remained the same. All these comparisons, summarized in Table 6, indicate that the Transformer-based encoder allows for parallel computation and thus faster generation with no degradation in the quality of the audio.

Considering these together, the findings validate that the proposed model outperforms the baseline in all dimensions, it generates speech that is more natural in quality, produces clearer and more intelligible output, and reduces runtime during inference. This sets up the Transformer-enhanced encoder as a strong, pragmatic candidate for high-quality Kurdish TTS applications.

Table 6: Performance comparison for both experiments (evaluated on 8-second audio samples).

Model	Inference Speed (s)	RTF
Baseline Model	0.596	0.0745
Proposed Model	0.431	0.0539

All the work behind Table 6, including training the model and measuring its speed, was performed under the same computational environment.

7 Discussion

The visualizations of attention alignment in Figures 6 and 7 indicate that the Transformer-based model proposed here generates sharper, more organized diagonal patterns than does the baseline CNN-BiLSTM architecture. This indicates more precise text-input and acoustic-output alignment, enabled by the long-range dependencies modeled in parallel through self-attention mechanism characteristic of Transformers. Such capabilities are especially advantageous for Kurdish, a morphologically rich and syntactically flexible language.

Training dynamics further reveal efficiency gains. The loss curves in Figures 4 and 5 show that, the Transformer-based encoder converged faster and with greater stability than the baseline model. These enhancements to loss reduction, along with enhanced performance with respect to both quality of attention alignment and training stability illustrates the advances made. A deeper model architecture was able to achieve faster convergence to generate higher quality speech than had also been achieved previously in a less stable manner during training.

Additionally, although our proposed system maintains an autoregressive characteristic of TTS and therefore maintains temporal consistency with respect to natural-sounding prosody, the implementation is accelerated significantly through the use of multiple parallelizable Transformer operations during the encoder portion of processing. In contrast to FastSpeech, which relies on a fully non-autoregressive model, there is a tradeoff to be made between better modelling of the sequence when speed is prioritized, as natural-sounding prosody is often sacrificed to achieve higher synthesis speed. Therefore, in our designs, we have created a hybrid system where autoregressive decoding is preserved for high-quality output while also leveraging the advantages of multiple parallel transformers for encoding.

The model's practicability was further validated through inference evaluation, achieving an RTF of 0.0539, yet the synthesis speed and memory use were still competitive with those of other TTSs and could be used in real-time applications in low-resource areas. Other TTS systems have shown lower performance when compared to our method. FastSpeech received a MOS of 3.84 and a recent Kurdish Transformer model [32] had an MOS of 3.94. However, our system achieved a 4.74 MOS, representing a new high-performance benchmark. This model utilized adversarial training and a variational module, while our model is more straightforward and efficient, thus resulting in a substantially higher level of performance. The statistically significant MOS improvement ($p < 0.001$) further confirms that this performance gain reflects a genuine enhancement in perceived speech naturalness rather than random listener variation.

This implies that it is possible, through selective architectural improvement in incorporating a Transformer encoder and the effective decoder of the baseline model, to achieve better outcomes than complete Transformer-based or adversarial pipelines, especially in the limited, clean, and well-separated data cases. In contrast to earlier transfer learning methodologies like [31], where English-pretrained models were used and suffered from script and phonetic mismatches, our model is trained from scratch on Kurdish-specific text and acoustic feature representations. This approach allows more precise phonetic alignment and more natural-sounding speech. The 0.80 MOS gain over earlier Kurdish systems affirms the advantages of language-specific modeling under low-resource settings. Although direct comparisons are limited due to dataset differences and lack of access to prior models, especially true for studies involving the Kurdish language, Table 7 summarizes related results alongside our own. While promising results are achieved, certain limitations remain. The system is now trained from a single-speaker corpus and is optimized for the Central Kurdish Sorani dialect. Future work can stretch to multi-speaker or cross-dialect modeling, explore control mechanisms for prosody (e.g., style tokens or pitch embeddings), and explore multilingual pretraining or semi-supervised learning. Such fronts may boost generalizability, expressiveness, and data efficiency. In summary, the proposed Transformer-enhanced model gains substantial improvements in speech quality and alignment efficiency and surpasses existing Kurdish TTS models across all key evaluation metrics. It lays a solid foundation for scalable and adaptable speech synthesis in other low-resource and morphologically complex languages.

Table 7: Summary of the results found in related works and proposed method.

Study	Method	Language	MOS
[28]	FastSpeech	English	3.84
[31]	Tacotron 2	Persian	3.01-3.97
[30]	Transfer learning tacotron 2	Arabic	4.21
[11]	Concatenative	Kurdish	Allophone 2.45 Syllable 3.02, Diphone 3.51
[33]	Transformer-based TTS with VAE and Adversarial Training	Kurdish	3.94
[32]	Transfer learning using pretrained model	Kurdish	4.10
Our Proposed Model	Transformer encoder with	Kurdish	4.74

	attention-based decoder		

Conclusion

This study developed a high-quality Central Kurdish TTS system built from scratch with a large-scale dataset (13.63 hours) containing 7,297 sentences spoken by a native adult male Sorani speaker. The dataset covers many topics and establishes a common baseline for model training and approach evaluation. The proposed model combines a Transformer-based encoder with an established attention-based decoder and is capable of learning long-range dependencies and prosodic variation in Kurdish. A WaveGlow vocoder was employed for reconstructing the waveform obtained from predicted mel-spectrograms, resulting in synthesized speech that was both natural and understandable. This proposed architecture outperformed the original baseline model on all of the evaluation metrics (MOS,WER, and RTF) tested. The synthesized speech was rated as more natural and intelligible, while inference remained computationally efficient, thereby making the system suitable for real-time applications. The architecture retains the autoregressive decoder to preserve natural prosody while enhancing the encoder using Transformer parallelism so that the quality is high and the synthesis is fast. This study highlights the efficacy of Transformer-based encoders in low-resource languages and set a new standard for Kurdish TTS. The results show that well-designed and high-quality data can be more effective than advanced or transfer-based methods, offering a workable solution for underrepresented, morphologically rich languages like Kurdish. In the future, advancing research can be achieved by incorporating multi-speaker and cross-dialectal modeling, and multilingual or semi-supervised training methods. Such advancements would further enrich generalizability and enhance the naturalness and adaptability of Kurdish speech synthesis in real-world usage.

References

[1] Hunt, A. J. & Black, A. W. Unit selection in a concatenative speech synthesis system using a large speech database. Proc ICASSP **1996**, 373-376 (1996).

[2] Tan, X., Qin, T., Soong, F. & Liu, T. Y. A survey on neural speech synthesis. arXiv preprint **2106**, 15561 (2021).

[3] Chen, F., Yang, J. & Zhao, L. A bilingual speech synthesis system of Standard Malay and Indonesian based on HMM DNN. Proc IALP **2020**, 181-186 (2020).

[4] Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y. & Skerry Ryan, R. J. Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions. *Proc ICASSP* **2018**, 4779-4783 (2018).

[5] Ren, Y., Hu, C., Tan, X., Qin, T., Zhao, S., Zhao, Z. & Liu, T. Y. FastSpeech 2 fast and high quality end to end text to speech. *arXiv preprint* **2006**, 04558 (2020).

[6] Ping, W., Peng, K., Gibiansky, A., Arik, S. O., Kannan, A., Narang, S., Raiman, J. & Miller, J. Deep Voice 3 scaling text to speech with convolutional sequence learning. *arXiv preprint* **1710**, 07654 (2017).

[7] Kong, J., Kim, J. & Bae, J. HiFi GAN generative adversarial networks for efficient and high fidelity speech synthesis. *Adv Neural Inf Process Syst* **33**, 17022-17033 (2020).

[8] Jabar, P. A comprehensive study of machine learning and deep learning methods for sentiment analysis on Kurdish Sorani text. *Passer J Basic Appl Sci* 7(2), 1118-1130 (2025).

[9] Fatemeh D. Enhancing Low-Resource Sentiment Analysis: A Transfer Learning Approach. Passer J Basic Appl Sci 6(2),265-274 (2025)

[10] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. Attention is all you need. *Adv Neural Inf Process Syst* **30**, 5998-6008 (2017).

[11] Klatt, D. H. Software for a cascade parallel formant synthesizer. *J Acoust Soc Am* **67**, 971-995 (1980).

[12] Birkholz, P. Modeling consonant vowel coarticulation for articulatory speech synthesis. *PLoS One* **8**, e60603 (2013).

[13] Bahrampour, A., Barkhoda, W. & Azami, B. Z. Implementation of three text to speech systems for Kurdish language. *Proc CIARP* **2009**, 321-328 (2009).

[14] Barkhoda, W., Zahir Azami, B., Bahrampour, A. & Shahryari, O. K. A comparison between allophone syllable and diphone based TTS systems for Kurdish language. *Proc ISSPIT* **2009**, 557-562 (2009).

[15] Daneshfar, F., Barkhoda, W. & Azami, B. Z. Implementation of a text to speech system for Kurdish language. *Proc ICDT* **2009**, 117-120 (2009).

[16] Hassani, H. & Kareem, R. Kurdish text to speech KTTS. *Proc IWIPS* **2011**, 79-89 (2011).

[17] Rao, K., Peng, F., Sak, H. & Beaufays, F. Grapheme to phoneme conversion using long short term memory recurrent neural networks. *Proc ICASSP* **2015**, 4225-4229 (2015).

[18] Yao, K. & Zweig, G. Sequence to sequence neural net models for grapheme to phoneme conversion. *arXiv preprint* **1506**, 00196 (2015).

[19] Ronanki, S., Henter, G. E., Wu, Z. & King, S. A template based approach for speech synthesis intonation generation using LSTMs. *Proc Interspeech* **2016**, 2463-2467 (2016).

[20] Zen, H. & Sak, H. Unidirectional long short term memory recurrent neural network with recurrent output layer for low latency speech synthesis. *Proc ICASSP* **2015**, 4470-4474 (2015).

[21] Mehri, S., Kumar, K., Gulrajani, I., Kumar, R., Jain, S., Sotelo, J., Courville, A. & Bengio, Y. SampleRNN an unconditional end to end neural audio generation model. *arXiv preprint* **1612**, 07837 (2016).

[22] Van Den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. & Kavukcuoglu, K. WaveNet a generative model for raw audio. *arXiv preprint* **1609**, 03499 (2016).

[23] Arik, S., Diamos, G., Gibiansky, A., Miller, J., Peng, K., Ping, W., Raiman, J. & Zhou, Y. Deep Voice 2 multi speaker neural text to speech. *arXiv preprint* **1705**, 08947 (2017).

[24] Wang, Y., Skerry Ryan, R. J., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z. & Bengio, S. Tacotron towards end to end speech synthesis. *arXiv preprint* **1703**, 10135 (2017).

[25] Sotelo, J., Mehri, S., Kumar, K., Santos, J. F., Kastner, K., Courville, A. & Bengio, Y. Char2Wav end to end speech synthesis. *arXiv preprint* **1702**, 07825 (2017).

[26] Prenger, R., Valle, R. & Catanzaro, B. WaveGlow a flow based generative network for speech synthesis. *Proc ICASSP* **2019**, 3617-3621 (2019).

[27] Ping, W., Peng, K. & Chen, J. Clarinet parallel wave generation in end to end text to speech. *arXiv preprint* **1807**, 07281 (2018).

[28] Ren, Y., Ruan, Y., Tan, X., Qin, T., Zhao, S., Zhao, Z. & Liu, T. Y. FastSpeech fast robust and controllable text to speech. *Adv Neural Inf Process Syst* **32**, 1-12 (2019).

[29] Win, Y., Lwin, H. P. & Masada, T. Myanmar text to speech system based on Tacotron end to end generative model. *Proc ICTC* **2020**, 572-577 (2020).

[30] Fahmy, F. K., Khalil, M. I. & Abbas, H. M. A transfer learning end to end Arabic text to speech deep architecture. *Proc ANNPR* **2020**, 266-277 (2020).

[31] Naderi, N., Nasersharif, B. & Nikoofard, A. Persian speech synthesis using enhanced Tacotron based on multi resolution convolution layers and a convex optimization method. *Multimedia Tools Appl* **81**, 3629-3645 (2022).

[32] Muhamad, S. & Veisi, H. End to end Kurdish speech synthesis based on transfer learning. *Passer J Basic Appl Sci* **4**, 150-160 (2022).

[33] Ahmad, H. A. & Rashid, T. A. Central Kurdish text to speech synthesis with novel end to end transformer training. *Algorithms* **17**, 292 (2024).

[34] Ahmad, H. A. & Rashid, T. A. Gigant KTTS dataset towards building an extensive gigant dataset for Kurdish text to speech systems. *Data Brief* **55**, 110753 (2024).