FastStitch: Speech editing by hitch-hiking a pre-trained FastSpeech2 model

Antonios Alexos* and Pierre Baldi

University of California, Irvine

Abstract

We present an innovative approach to speech editing, mitigating the time-consuming process of training acoustic models from scratch. Our methodology involves fine-tuning the upper layers of a pre-trained FastSpeech2 model and fusing it with information from a reference mel-spectrogram during inference via a convolution-based, or an attention-based, blending network. Comparative evaluations against baseline methods and against state-of-the-art techniques on single-speaker (LJSpeech) as well as multi-speaker (VCTK) datasets, employing both subjective and objective measures, demonstrate the superior quality of our approach, yielding significantly more natural-sounding speech edits[1].

1 Introduction

Speech synthesis technology has evolved significantly since the rise of deep learning, originally developed to emulate human speech production. What were once robotic-sounding synthesized voices have now become ubiquitous, gracing call centers, smartphone applications, and virtual avatars. This remarkable progress is a direct response to the surging prevalence of speech-driven interactions in our increasingly sophisticated personal devices. Notably, in the past decade, we have witnessed a substantial improvement in speech synthesis quality, drawing ever closer to human-like attributes, all thanks to the advancements in end-to-end neural synthesis.

As the realism of synthetic voices continues to advance, an array of applications beyond traditional text-to-speech synthesis has emerged, encompassing voice conversion [1], accessibility-driven cloning [2], expressive synthesis [3], and speech editing [4]. Many of these applications necessitate the generation of fully synthetic content, introducing intricate challenges in navigating the uncanny valley and balancing the interplay between intelligibility and naturalness. These comprehensive synthesis tasks are often denoted as complete synthesis approaches. In contrast, the domain of speech editing entails synthesizing segments that are juxtaposed with reference recordings, subjecting the synthesized content to direct comparison by listeners within its original context. This unique characteristic heightens the perceptual complexity, as listeners can readily assess synthesis quality in relation to the original, leading to its categorization as a partial synthesis task.

The formal definition of a speech editing task involves modifying the audio content of a reference speech sample R and its transcript T_r to create a transformed transcript T_e." The word count in T_r and T_e can vary, but the goal is to ensure that the replaced segment maintains similar voice quality, while the overall edit sounds seamless and natural. Speech editing is valuable in scenarios requiring repetitive playback of content with slight variations, such as public announcements or customized messages. It allows for altering an expressive voiceover to suit specific contexts, preserving much of the original voice quality, which can be challenging with a comprehensive speech synthesis system.

The research community has explored various speech editing approaches. One method integrates segments from the same speaker using pitch and prosody features for natural editing [5]. Another approach generates audio in a generic voice and converts it to the desired target voice [6], but has noticeable roughness at edit boundaries. EditSpeech [4] uses forward and backward decoders for fused mel-spectrograms, while A^{3}T [7] introduces cross-modal alignment embedding. EdiTTS [8] refines edited speech with perturbations to Gaussian priors. SpeechPainter [9] fills speech gaps using an attention-based model, limited to 1-second gaps. MaskedSpeech [10] focuses on Mandarin speech editing with a pretrained FastSpeech2 model. Our novel approach expedites training and achieves smoother audio segment integration by incorporating an auxiliary module into a pretrained Text-to-Speech model like FastSpeech2 (FS2), enhancing efficiency and naturalness in audio stitching for broader applications.

Our main contributions are as follows:

- We introduce an innovative approach that accelerates speech-editing network training by leveraging a pre-trained FS2 model.
- We propose two specialized auxiliary modules for fast and high-quality synthesis with automated editing capabilities: a convolution-based blending network for single-speaker data and

*Corresponding Author: aalexos@uci.edu

[1] Audio samples can be found webpage or gdrive link.

Proceedings of the 5th Northern Lights Deep Learning Conference (NLDL), PMLR 233, 2024.

an attention-based blending network for multi-speaker data.

- Through comprehensive subjective and objective tests, we demonstrate the superiority of our approach over baseline methods and state-of-the-art approaches.

2 Method

2.1 Motivation

Our choice to build upon the FS2 model stems from its remarkable success within the domain of TTS models. FS2 offers significant improvements, particularly in mitigating challenges associated with non-autoregressive TTS approaches. Notably, FS2 incorporates crucial modules such as duration prediction for phonemes and the incorporation of variation information related to speech attributes like pitch and energy. FS2 embraces the utilization of ground-truth targets during training, which enhances audio quality while preventing information loss. These inherent advantages and the robust architecture of FS2 render it an ideal foundation upon which to integrate our proposed module, allowing us to harness the model's strengths.

Our approach, named "FastStitch," augments the FS2 model with an auxiliary module, convolution-based or attention-based, enabling the seamless integration of synthesized words into recorded audio. During training, we employ masked mel-spectrograms, akin to [7], and in inference, we adaptively adjust source masks to match predicted word durations, ensuring natural and coherent output. This innovative combination of components enhances the FS2 model for efficient speech editing.

2.2 Blending Network

The core element within FastStitch, facilitating speech editing, is the blending network. We introduce two distinct configurations for this network: a convolution-based blending network tailored for single-speaker data and a more advanced attention-based blending network designed for multi-speaker data scenarios. This strategic choice ensures optimal performance in different contexts.

Convolution Blending Network. The blending network in FastStitch shares a structural resemblance with the post-net of FS2, featuring six 2D convolutions with a kernel size of 5. It concatenates the predicted mel-spectrogram (S_p) and the masked reference mel-spectrogram (S_e) along channels. These concatenated mel-spectrograms undergo cross-convolution and pass through a sigmoid mask to effectively combine them, facilitating the seamless integration of synthesized and reference audio segments in speech editing.

Attention Blending Network. It employs a double attention block [11], originally designed for global feature propagation in images, to efficiently facilitate access to these features by the rest of the model. This block operates in two sequential steps: first, it gathers image features using an attention-pooling operation, and second, it selects and distributes these features through attention mechanisms. In our context, this double attention block collects features from the synthesized mel-spectrogram to populate the masked regions of the reference mel-spectrogram corresponding to the edited speech segments. It's worth noting that, in contrast to the convolution-based blending network, the attention-based blending network does not require a sigmoid mask for blending; instead, attention mechanisms fulfill this role.

2.3 Spectrogram blending

We initiate our approach by pre-training the FS2 model on a designated speaker dataset, such as LJSpeech [12] and VCTK [13]. The core FS2 model remains unchanged up to the length regulator and decoder stages. We then position the blending network between the decoder and the post-net of the primary FS2 model. This blending network serves the critical function of incorporating information from a masked reference mel-spectrogram into the intermediate output generated by the decoder, as depicted in Figure 1 (reference to the visual representation is available in the original paper).

Our hypothesis posits that during fine-tuning, exposing the auxiliary module to a partial reference mel-spectrogram enables the network to effectively utilize the text input to fill in the masked regions, while seamlessly incorporating the unmasked segments from the pre-recorded context. Notably, our approach to training the auxiliary network exhibits notable efficiency compared to [7], primarily because the primary FS2 model has already acquired the capacity to synthesize fluent speech from text. Our empirical findings underscore the pivotal role of mask selection in speech editing, significantly influencing convergence rates and overall stability. More comprehensive details regarding the mask are elaborated upon in Section 3.

At inference time, to edit an utterance we first force-align it using a pretrained Kaldi [14] hybrid acoustic model that is trained from scratch on the train data. The phone-level alignment is used to determine the word boundary for the word(s) to be replaced. The corresponding portion of the mel-spectrogram is masked according to one of the strategies described in section 3. The reference text T_r is also modified by replacing the word(s) to obtain the modified text T_e. Note that T_e may contain a different number of words from T_r. We then forward pass the phonemes in T_e through the FS2 main network past the length regulator before the penultimate post-net to obtain the mel-spectrogram predicted by the acoustic model, S_p. The total duration predicted by the VariancePredictor [15] is then used to resize the length of our mask in the reference mel-spectrogram S_e so that it matches the length of S_p. The main operation of speech editing happens in the blending network, and it is necessary to align the predicted and masked mel-spectrograms as described in section 2.2. The blended mel-spectrogram is then passed through the post-net which is also fine-tuned, followed by a pretrained and finetuned Hifi-GAN vocoder. The architecture of the proposed methodology is depicted in fig. 1, where we see the frozen and the unfrozen parts of the network, as well as the proposed blending network with the speech editing operation.

During the speech editing inference phase, we apply mel-spectrogram masking based on strategies outlined in section 3. Simultaneously, we modify reference text T_r to create T_e. Phonemes from T_e pass through the primary FS2 network beyond the length regulator, yielding a mel-spectrogram prediction S_p. We use VariancePredictor [15] duration predictions to resize the reference mel-spectrogram mask (S_e) for alignment with S_p in case of size indifference. Speech editing primarily occurs in the blending network (see section 2.2), followed by the post-net for fine-tuning, and a pretrained, fine-tuned Hifi-GAN vocoder for generating edited speech. fig. 1 visually outlines our methodology's structure, the frozen and unfrozen segments of the network, emphasizing the blending network's significance in the speech editing process.

3 Experiments

This section outlines our experimental setup for evaluating the proposed method. As our approach extends the FS2 network, we adhere to the FS2 framework in terms of audio sampling, mel-spectrogram channels, and forced alignment. We conducted our experiments using the PyTorch implementation of FS2, available at [16]. Notably, this implementation includes a post-net, a commonly employed component for fine-tuning the output mel-spectrogram. In addition to the loss function from FS2, we introduce two Mean Absolute Error (MAE) terms: one between S_f and S_e to expedite the blending network's convergence, and another between the post-net outputs, contributing to overall refinement.

In our experimental setup for both LJSpeech and VCTK datasets, we initiate pre-training of the standard FS2 network, extending over 200k training steps. We employ a random train-validation split and employ the subjective evaluation study, which we adapt from EditSpeech and A^{3}T. It is important to note that we ensure the exclusion of any validation samples from our training set for this study. Subsequently, following the outlined steps in section 2.3, we freeze the FS2 network up to the decoder and proceed to train all layers above the blending network. Additionally, we observe that extending the pre-training of the FS2 network to 900k steps may impede the convergence of the auxiliary network. Therefore, we opt to utilize an earlier checkpoint for fine-tuning, facilitating smoother convergence of the auxiliary network.

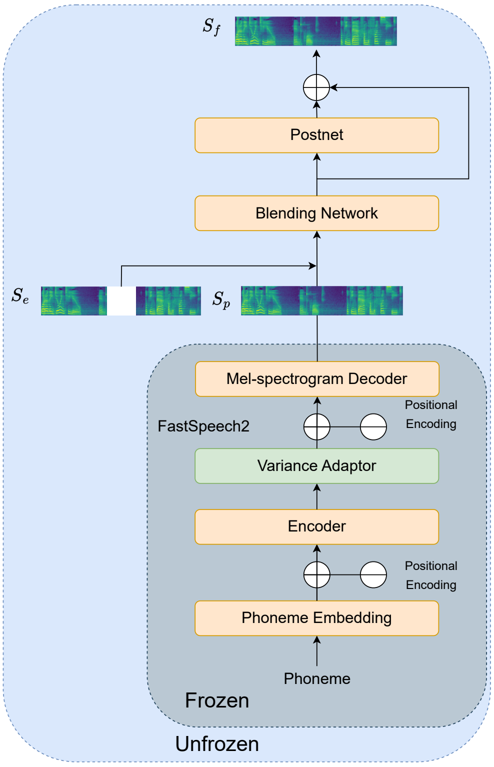

Figure 1. The FastStitch model, which consists of the FS2 model along with the proposed blending network, and post-net. The blending network can either be convolution-based or attention-based.

During the training phase, we incorporate random masking, covering approximately 10% of the reference mel-spectrogram's central region with zeros. Notably, this procedure remains equally effective if applied to either the beginning or end of the spectrogram. We also explore replacing the masked region with gaussian noise which leads to slower convergence and less seamless blending of mel-spectrograms. Following the masking of the reference mel-spectrogram, we concatenate it with the synthesized mel-spectrogram, inputting this combined data into the blending network. Additionally, we experiment with an alternative approach, attempting to predict the fused mel-spectrogram directly through 2D convolutions and pooling operations, bypassing the proposed blending network. However, this alternative method proves ineffective, as the model struggles to learn the final, edited mel-spectrogram for speech.

During inference, we apply mel-spectrogram masking based on the edited words, using our knowledge of their phoneme boundaries. This masking procedure replicates the training methodology by replacing the relevant segment of the mel-spectrogram with zeros. Concurrently, we substitute the word(s) in the reference text with the corresponding target words. To ensure alignment, we dynamically resize the mask according to FS2's duration predictor, ensuring that both the reference mel-spectrogram and the edited mel-spectrogram maintain identical lengths.

3.1 Baselines

In addition to our proposed speech editing approach utilizing the blending network, we introduce two alternative baseline methods for comparison in the single-speaker setup. These approaches operate under the assumption that we possess the reference audio R, which is force-aligned to the original text T_r, thereby making the boundaries of the source word(s) already known. Furthermore, these methods do not necessitate any additional fine-tuning beyond the utilization of a pretrained complete synthesis model.

Complete synthesis and swap: We first use a pretrained FS2 model to synthesize speech corresponding to the edited text T_e. We predict the durations from a pretrained FS2 and use them to find the word boundaries. Then we replace the source word(s) in the reference audio with the synthesized targeted word(s).

FeatSwitch: In FeatSwitch we switch prosody features in FS2 mid-inference to achieve speech editing. We first extract phoneme-level energy, pitch, and duration features from the reference audio R. Then we predict the same features from the edited text T_e with FS2. Finally, we replace back, ground truth features into all phonemes that don't belong to the target word(s) in the synthesized edited audio. This altered feature sequence is then fed through the rest of the network to obtain a speech edited representation.

In the context of the single-speaker setup, we opt not to include a comparison to fully resynthesized edited text, since the final audio sample can still exhibit differences compared to the reference. Conversely, for multi-speaker data, we choose to benchmark FastStitch against state-of-the-art speech editing methodologies such as A^{3}T and EditSpeech. Although alternative approaches like SpeechPainter, EdiTTS, and MaskedSpeech exist, the distinct nature of these methods renders them challenging to directly compare to FastStitch. SpeechPainter, for instance, primarily fills short-time gaps in audio samples with identical audio content and does not perform speech editing. EdiTTS is primarily evaluated on single-speaker data, and MaskedSpeech operates exclusively at the sentence level within the Mandarin language.

4 Evaluation

In the evaluation of speech synthesis and speech editing methods, both subjective and objective assessments are commonly employed. Objective evaluation typically utilizes metrics like the Mel-cepstral distance (MCD) [17] to measure audio dissimilarity, while subjective evaluations involve perceptual listening tests, often using mean opinion scores (MOS) [18] to gauge human perception. In recent years, neural network-based learned evaluation scores have gained traction, addressing the limitations of small population sizes in subjective tests (typically $N \approx 15$). This practice, prevalent in the TTS community, has been extended to speech editing research, where studies like [7] and [4] use MOS for edited samples to compare their methods to baselines, enhancing the evaluation robustness.

To validate our approach, we conducted a preliminary experiment using the LJSpeech dataset, involving subjective assessment of naturalness. Three random samples from the LJSpeech validation split were presented to human subjects, alongside the reference audio (R), FastStitch, and the baselines in randomized order for naturalness rating. We then replicated a similar setup on the VCTK dataset, including the reference (R) and samples from the state-of-the-art approaches, EditSpeech and A^{3}T, from their official websites as discussed in section 3.1. We carefully selected four samples for each method, maintaining consistency across all methods. In the multi-speaker subjective evaluation, subjects rated the overall MOS for identical edited sentences from three different sources.

For LJSpeech's objective evaluation, we randomly selected 32 uniformly distributed samples from the validation split. In contrast, for VCTK's objective evaluation, we used the same samples employed in our subjective assessment. Our experiments with FastStitch encompassed the convolution-based blending network for LJSpeech and the attention-based blending network for VCTK. Notably, our comparative analysis revealed that the convolution-based blending network's performance lagged behind the attention-based counterpart, particularly in the more intricate multi-speaker VCTK dataset. This discrepancy in performance can be attributed to VCTK's heightened complexity due to limited data for each speaker, in contrast to the data-rich LJSpeech single-speaker setup.

4.1 Results and Discussion

MOS ratings were collected for edited audio samples using a subjective study involving 15 participants who assessed samples from both the LJSpeech and VCTK datasets. This study comprised two distinct segments, one dedicated to each dataset. Participants were asked to express their judgments on a 1-5 Likert scale [19]. Notably, in both segments, we thoughtfully incorporated reference samples alongside the edited ones. This inclusion served dual purposes: first, it established a baseline for assessing the naturalness and overall speech quality, and second, it provided an opportunity for subjects to potentially misidentify unedited samples as edited ones. Detailed results of the evaluation study on LJSpeech are presented in section 4.1.1, while section 4.1.2 showcases the outcomes pertaining to the VCTK dataset.

4.1.1 LJSpeech

For the subjective evaluation of LJSpeech, participants were instructed to assess the naturalness of audio samples. The convolution-based FastStitch achieved a MOS of 3.8, while the reference audio samples garnered a notably higher mean score of 4.86, but FastStitch still outperformed the baseline methods. In addition to the subjective evaluation, we conducted an objective assessment by calculating the MCD for FastStitch on LJSpeech, resulting in a value of 4.98 over 32 synthesized samples. The amalgamation of subjective and objective evaluation scores in the context of LJSpeech underscores FastStitch's commendable performance in the realm of single-speaker data. The subjective results are visually presented in the box plot shown in fig. 2. Furthermore, our qualitative analysis of FastStitch's performance on single-speaker data, coupled with MCD scores from the held-out validation set, revealed an interesting pattern: samples with lower MCD values (≤ 8) exhibited smoother and more natural edits, whereas those with higher MCD values (≥ 8) often exhibited perceptible artifacts at word boundaries. We hypothesize that this phenomenon may be attributed to duration prediction errors in FS2, as suggested by the high MCD scores, since the current masking before blending network relies on predicted durations.

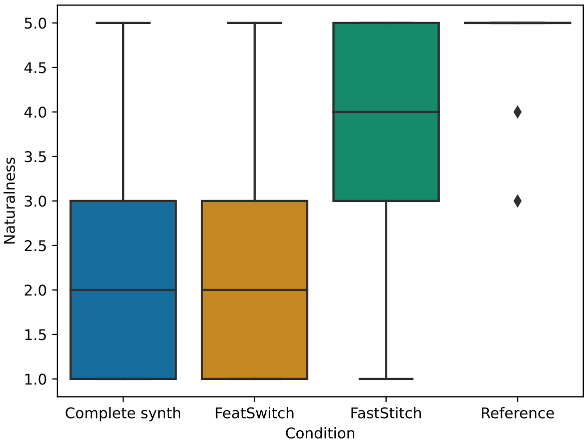

Figure 2. Box plot of naturalness ratings for the experimental conditions and reference (unedited) audio. Our proposed method, FastStitch, significantly outperforms both of the baseline conditions.

4.1.2 VCTK

In our multi-speaker evaluation on the VCTK dataset, participants were tasked with rating the naturalness of audio samples, where the attention-based FastStitch achieved a MOS of 3.51, outperforming both EditSpeech (MOS 3.28) and A^{3}T (MOS 3.3), while the reference audio garnered a MOS of 4.43. The VCTK dataset's diverse speaker population, encompassing a variety of accents with limited data for each speaker, contributed to relatively lower MOS scores compared to LJSpeech. In terms of objective evaluation using MCD scores, our method consistently surpassed state-of-the-art approaches, as detailed in table 1, providing a comprehensive overview of the assessment results.

Method	**MOS (↑)**	**MCD (↓)**
EditSpeech	3.28±0.33	7.54
A^{3}T	3.3±0.35	7.97
FastStitch	**3.51±0.23**	**6.5**
Reference	4.43±0.2	-

Table 1. VCTK

Table 2. MOS (↑) and MCD (↓) scores for FastStitch, the compared methods and the reference samples with 95% confidence intervals for LJSpeech and VCTK. FastStitch outperforms the compared methods in both metrics.

5 Conclusions

In our work, we introduce FastStitch, an innovative approach to enhance a pretrained FS2 model for speech editing. It incorporates two blending networks tailored for single-speaker and multi-speaker data. FastStitch outperforms both baselines and state-of-the-art methods in speech editing.

References

[1] S. H. Mohammadi and A. Kain. “An overview of voice conversion systems”. In: *Speech Communication* 88 (2017), pp. 65–82.

[2] Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Z. Chen, R. Skerry-Ryan, Y. Jia, A. Rosenberg, and B. Ramabhadran. “Learning to speak fluently in a foreign language: Multilingual speech synthesis and cross-language voice cloning”. In: *arXiv preprint arXiv:1907.04448* (2019).

[3] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. Weiss, R. Clark, and R. A. Saurous. “Towards end-to-end prosody transfer for expressive speech synthesis with tacotron”. In: *international conference on machine learning*. PMLR. 2018, pp. 4693–4702.

[4] D. Tan, L. Deng, Y. T. Yeung, X. Jiang, X. Chen, and T. Lee. “Editspeech: A text based speech editing system using partial inference and bidirectional fusion”. In: *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE. 2021, pp. 626–633.

[5] M. Morrison, L. Rencker, Z. Jin, N. J. Bryan, J.-P. Caceres, and B. Pardo. “Context-aware prosody correction for text-based speech editing”. In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 7038–7042.

[6] Z. Jin, G. J. Mysore, S. Diverdi, J. Lu, and A. Finkelstein. “Voco: Text-based insertion and replacement in audio narration”. In: *ACM Transactions on Graphics (TOG)* 36.4 (2017), pp. 1–13.

[7] H. Bai, R. Zheng, J. Chen, M. Ma, X. Li, and L. Huang. “A3T: Alignment-Aware Acoustic and Text Pretraining for Speech Synthesis and Editing”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 1399–1411.

[8] J. Tae, H. Kim, and T. Kim. “EdiTTS: Score-based Editing for Controllable Text-to-Speech”. In: *arXiv preprint arXiv:2110.02584* (2021).

[9] Z. Borsos, M. Sharifi, and M. Tagliasacchi. “SpeechPainter: Text-conditioned Speech Inpainting”. In: *arXiv preprint arXiv:2202.07273* (2022).

[10] Y.-J. Zhang, W. Song, Y. Yue, Z. Zhang, Y. Wu, and X. He. “MaskedSpeech: Context-aware Speech Synthesis with Masking Strategy”. In: *arXiv preprint arXiv:2211.06170* (2022).

[11] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng. “Aˆ 2-nets: Double attention networks”. In: *Advances in neural information processing systems* 31 (2018).

[12] K. Ito and L. Johnson. *The LJ Speech Dataset*. https://keithito.com/LJ-Speech-Dataset/. 2017.

[13] J. Yamagishi, T. Arai, M. Dong, S. King, S. R. Chávez, S. E. King, J. D. O’Shea, K. Oura, H. Kawai, and J. Zhang. “CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit (version 0.92)”. In: *Proc. of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. Portorož, Slovenia, 2019, pp. 3636–3642.

[14] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely. “The Kaldi Speech Recognition Toolkit”. In: *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Catalog No.: CFP11SRW-USB. Hilton Waikoloa Village, Big Island, Hawaii, US: IEEE Signal Processing Society, Dec. 2011.

[15] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu. “Fastspeech 2: Fast and high-quality end-to-end text to speech”. In: *International Conference on Learning Representations (ICLR) 2021* (2020).

[16] C.-M. Chien. *FastSpeech 2 - PyTorch Implementation*. https://github.com/ming024/FastSpeech2. 2021.

[17] 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. IEEE. 1993, pp. 125–128.

[18] ITU-TRecommendationP.10. *Vocabulary for performance, quality of service and quality of experience, Geneva: International Telecommunication Union*. 2017.

[19] A. Joshi, S. Kale, S. Chandel, and D. K. Pal. “Likert scale: Explored and explained”. In: *British journal of applied science & technology* 7.4 (2015), p. 396.