POLITECNICO DI TORINO
Repository ISTITUZIONALE

A Deep Learning Approach for Low-Latency Packet Loss Concealment of Audio Signals in Networked Music Performance Applications

Original
A Deep Learning Approach for Low-Latency Packet Loss Concealment of Audio Signals in Networked Music Performance Applications / Verma, P.; Mezzay, A. I.; Chafe, C.; Rottondi, C.. - ELETTRONICO. - 2020-:(2020), pp. 268-275. (Intervento presentato al convegno 27th Conference of Open Innovations Association FRUCT, FRUCT 2020 tenutosi a Trento, Italia nel 2020) [10.23919/FRUCT49677.2020.9210988].

Availability:
This version is available at: 11583/2850651 since: 2020-10-30T16:46:47Z

Publisher:
IEEE Computer Society

Published
DOI:10.23919/FRUCT49677.2020.9210988

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

25 April 2024

A Deep Learning Approach for Low-Latency Packet Loss Concealment of Audio Signals in Networked Music Performance Applications

Prateek Verma*, Alessandro Ilic Mezza†, Chris Chafe*, Cristina Rottondi‡
* Stanford University, Stanford, California
† Politecnico di Milano, Milan, Italy
‡ Politecnico di Torino, Turin, Italy
{prateekv, cc}@ccrma.Stanford.edu, alessandroilic.mezza@polimi.it, cristina.rottondi@polito.it

***Abstract*—Networked Music Performance (NMP) is envisioned as a potential game changer among Internet applications: it aims at revolutionizing the traditional concept of musical interaction by enabling remote musicians to interact and perform together through a telecommunication network. Ensuring realistic conditions for music performance, however, constitutes a significant engineering challenge due to extremely strict requirements in terms of audio quality and, most importantly, network delay. To minimize the end-to-end delay experienced by the musicians, typical implementations of NMP applications use uncompressed, bidirectional audio streams and leverage UDP as transport protocol. Being connectionless and unreliable, audio packets transmitted via UDP which become lost in transit are not retransmitted and thus cause glitches in the receiver audio playout. This article describes a technique for predicting lost packet content in real-time using a deep learning approach. The ability of concealing errors in real time can help mitigate audio impairments caused by packet losses, thus improving the quality of audio playout in real-world scenarios.**

I. Introduction

Performing music together at distances across the Internet is a practice that makes use of low-latency uncompressed audio streaming technologies. According to several studies [1], [2], the delay tolerance threshold is estimated to be 20–30 ms, corresponding to a distance of approximately 8–9 m (considering the speed of sound propagation in air), a physical spacing which can be considered as a maximum separation ensuring the maintenance of a common tempo without a conductor. From the networking point of view, very strict requirements must therefore be satisfied to keep the one-way end-to-end transmission delay below a few tens of milliseconds.

A common solution sends UDP audio packets directly between a pair of network hosts (full-duplex, peer-to-peer), thus avoiding the processing time required by compression codecs and delays introduced by the packet re-transmission mechanisms supported by TCP, while sacrificing data transfer reliability. Despite having no guarantees that the audio exchange will be error-free, good networks and appropriate tuning of audio and network parameters can still result in connections which are quite satisfactory for music performance, and even for concert presentations. Experiences with distributed ensembles [3] have included multi-channel audio and multi-site collaborations (greater than stereo and greater than two sites).

In order to ensure optimal conditions for ensemble synchronization, a strict deadline is imposed on the arrival of each audio packet at the receiving site. The amount of time between when the packet was generated on the sender side and this playback deadline constitutes part of the time delay or lag experienced by the collaborating musicians. Added to this is any propagation delay in air such as between instrument or voice and microphone, and loudspeaker to ear, as well as time spent in signal conversion between analog and digital. Packet streams transmitted across a Wide Area Network (WAN) require buffering at the receiver to account for jitter (i.e., variations of packet inter-arrival times). Critically, the buffer queue must be kept as low as possible since longer "cushions" (longer buffer queues) create longer transmission delays. A lost packet causes a gap in the playback audio, and an audio packet that arrives so late that the buffer queue has been exhausted is effectively a lost packet. Since the audio signal is not encoded prior

to transmission, the receiving side cannot rely on error correction mechanisms implemented in state-of-the-art audio codecs. Therefore, alternative recovery algorithms must be devised to cope with packet losses, in order to mitigate the resulting audio glitches. Such solutions must operate in real time and without introducing additional processing delays that would worsen the end-to-end latency.

In the reminder of this paper, we will focus on jacktrip [4], which is an open-source project used for concert and online jamming. The jacktrip application belongs to a class of uncompressed audio streaming applications including Soundjack [5], LOLA [6] and UltraGrid [7]. If a packet is lost in a jacktrip connection, its built-in Packet Loss Concealment (PLC) algorithm simply repeats the last good packet that was received. Alternatively, its PLC mode can be set to mute the audio until the next good packet is able to be played out. In most cases, these methods of coping with signal errors are inappropriate and some degree of distortion is audible.

What to do to repair the gaps from lost packets is the subject of this paper, which proposes a method to predict the missing audio data in real-time by leveraging a deep learning framework. The Machine Learning (ML) algorithm which will be described is trained with audio samples extracted from a database of music which resembles the music that will be transmitted. The effectiveness of the proposed solution is evaluated by artificially introducing gaps in a test audio signal, inserting intentional errors which mimic in a controlled way the packet losses due to transmission through a WAN. The test algorithm predicts the missing audio and its performance is compared to a state-of-the-art AutoRegressive (AR) model used for PLC. Results show that, for a range of configuration parameters, the ML algorithm outperforms the AR approach in repairing audio signal gaps.

The remainder of this paper is organized as follows: after briefly reviewing the related literature in Section II, we provide some background notions on AR models in Section III and then describe the proposed ML framework in Section IV. A performance assessment is provided in Section V.

II. Related Work

NMP over packet switched internets was first experimented with in the 90's and has gathered increasing attention in the scientific community in the last two decades. The widespread diffusion of internet services and their continuous performance improvements in terms of latency and capacity have fostered a music-ensembles-at-a-distance performance practice which is particularly timely at the time of this writing while the world copes with the SARS-CoV-2 pandemic and social distancing is required. For a thorough overview on NMP technologies, the reader is referred to [8]. Due to the best-effort paradigm of the IP protocol and wide differences in internet provisioning, packet jitter and losses due to network congestion or excessively delayed transmissions are generally unavoidable and require some form of mitigation. PLC methods for real-time multimedia streaming have been widely studied [9] and recently ML approaches for speech reconstruction using neural networks have been proposed [10], [11].

Of the studies focused on NMP applications, the majority treat error recovery for transmission of MIDI signals. The authors of [12] propose a PLC approach that relies on the RTP protocol and its associated transport control protocol RTCP. The solution leverages a recovery journal section within the RTP packet which holds information a receiver uses to recover from earlier lost packets. This way, the first packet received after one or multiple consecutive packet losses enables the recovery from all artifacts caused by missing audio data. The drawback of this approach is that it generates transmission overheads. Other recovery approaches for MIDI signals rely on auxiliary reliably-connected channels to transmit critical MIDI events [13] or implement acknowledgment mechanisms to allow for re-transmission of lost packets [14].

In [15], a method for the detection of beat patterns of music signals is discussed as a means for prediction. At the sender side, the beat information is included as ancillary data to a preceding audio data interval in the transmitted compressed audio stream and is leveraged to perform PLC at the receiver. The method incurs transmission overhead. The authors of [16] propose a low-cost period extraction and alignment module to synthesize concealment signals from previous raw audio data, thus avoiding the use of autoregressive models in order to reduce the computation time. Period extraction is based on zero-crossings and matched pre-processing. To ensure smooth transitions, the extrapolated blocks are cross-faded from/into the previous/following audio frames. The main difference between the method proposed in this paper and the one described in [16] is that, instead of repeating periods extracted from the previously received signal, we synthesize the waveform to be inserted in

place of a missing packet. There has also been prior work which incorporates time scale modification techniques for audio concealment [17]. The authors of [17] compared their method with existing techniques (silence substitution and pattern replication). A next wave of PLC techniques based upon AR models were shown to outperform these approaches [18]. These methods are similar to infill approaches in the image domain [19] which try to predict raw values of missing pixels based upon the neighboring content of an image.

The approach described here looks at both the spectrogram and the waveform of the input to predict the upcoming time-domain signal within a short interval. The method makes use of a latent space representation which summarizes the content of the signal in order to predict the next audio packet. The advantage of using latent space-based approaches is that a suitable summary representation can be learned according to the problem of interest, e.g., to simultaneously reconstruct a speech signal and to decode its transcription [20] or to transform audio signals from a given domain to another domain [21]. Additionally, latent representations have been used to reconstruct the signal of interest in problems such as style transfer [22] and voice transformation [23] by successfully mapping contents and characteristics of audio representations onto lower-dimensional vectors.

III. Background on Autoregressive Prediction Models

AutoRegressive (AR) models of stochastic processes are described by a set of parameters which are the coefficients of the linear regression of the current output random variable against its previous values. By assuming such a model for the transmitted audio signal, an AR model of order p, denoted by AR(p), can be effectively used to forecast the future behavior of the time series as a linear combination of its past p samples. The regression parameters can be computed and dynamically updated using the most recently received packets (usually the last valid buffer) by, for example, solving an Ordinary Least Square (OLS) problem or as the solution of the well-known Yule-Walker equations. Since speech signals are often quasi-stationary within short-time analysis windows, AR models have been found to be useful in predicting the content of missing packets [18], [24]. Similarly, music signals are generally well behaved and analogous assumptions can be made, especially in the case of recordings of a single pitched instrument. Therefore, we use this simple approach as a baseline to evaluate the performance of our ML model. Notice, however, that various extensions to the classical AR model have been proposed in the literature, such as the inclusion of a long-term predictor whose lag is tuned according to the estimated pitch. For a detailed overview of AR models and other signal processing approaches, readers are advised to refer to [25].

IV. Deep Learning Framework for Low-Latency Audio Prediction

Modeling longer term dependencies is considered a hard ML problem, especially for time-domain audio signals [26]. In the literature, a similar problem has been addressed in the field of computer vision, when trying to reconstruct a missing part of an image [19]. The high sample rates of digital audio signals present a significant challenge for learning small compact representations of long sequences, even for durations of 2–3 seconds.

We adopt an hybrid approach which makes the problem tractable by combining both spectral and time-domain signal representations. The hypothesis of our approach is as follows: for a missing packet, the correctly predicted waveform will depend on what has happened in the past as well as the smoothness of the reconstructed signal. The latter requirement refers to the continuity of the signal derivatives at the edges between subsequent packets which need to be smooth in order to avoid audible clicks or glitches. As pointed out in [26], it is difficult to model waveforms on the scale of a few seconds, as these can be time sequences on the order of 50k samples or greater (considering 16 kHz as the sampling rate). In order to learn the behavior of time-domain signals characterized by longer-term dependencies and predict packets to come, we utilize spectro-temporal representations of waveforms in a training database. The same transform is applied to already received packets which are stored in a first-in first-out buffer. We compute a 100-bin mel-spectrogram of the previous 2 s of audio using a hop size of 10 ms and a 30 ms long Hanning window. A representation frame of 100×200 corresponds to the past 2 s of audio. We also extended the previous context to 4 s and 8 s and sub-sample the mel-spectrograms by a factor of 2 and 4, respectively, to yield the same representation frame size. All three frames (2 s, 4 s and 8 s) are stacked into 3 channels of a real-valued tensor of size $100\times200\times3$. The mel-spectrogram is magnitude only (which throws away the phase/relative shift of the time-domain signal). Our assumption is that only the last valid packet matters when we synthesize the missing

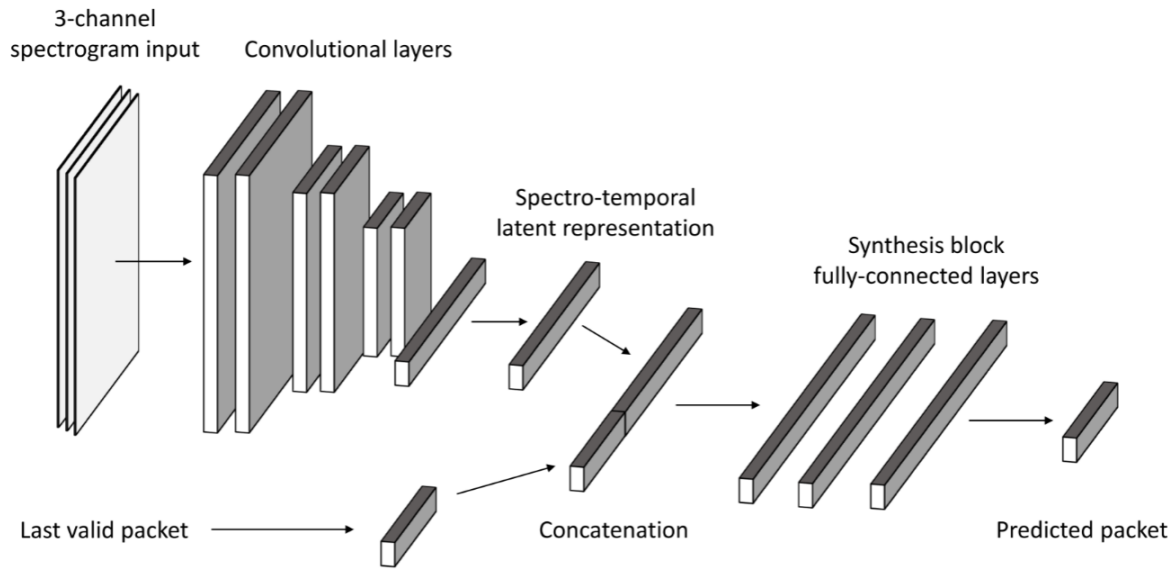

Fig. 1. Architecture of the proposed method for synthesizing new audio packets using a neural network.

future packet in order to ensure the smoothness of the reconstructed waveform.

The first part of our hybrid approach uses a convolutional neural network which takes as input the $100\times200\times3$ spectro-temporal representation and produces a 512-dimensional vector. This greater than 100-fold dimensional reduction proceeds as follows: a 3-layer convolutional architecture with 3×3 kernels and 1×2 pooling is connected to 2×2 pooling in the next two layers and finally a linear layer reduces the dimensionality to 512. Each of the convolutional layers has 64 channels.

The second part of our hybrid approach synthesizes a valid packet, conditioned on the latent code as extracted from the past signal history by the convolutional neural network. It concatenates the immediately previous valid packet of size 128 with the latent code of the past spectro-temporal content of size 512. These two signals, one for learning continuity in the time domain and one for the spectral content, are concatenated to produce a summary vector which contains both aspects of the signal. This is fed to a synthesis neural network, consisting of 3 fully-connected layers of 1024 neurons, and which is used to predict the missing or lost packet as depicted in Fig. 1.

We minimize the L_1 norm of the difference between the predicted packet and the target one. The latent summary vector, encapsulating the spectral content of the past, along with the synthesis block that generates the missing packet, are all learn-able from the data by adjusting the network weights via backpropagation. The Adam optimizer [27] was used for the network training with learning rate decay starting from 10^{-3} and then lowered to 10^{-7} over the course of 100 epochs . The L_1 norm was seen to have better convergence than the L_2 norm, one of the reasons being that all of the signals are in the range of $[-1, 1]$.

Not every audio sample point is equal with regards to error when stitching the predicted packet into an audio stream. In terms of continuity, errors at the beginning and at the end of a packet are much more perceptually significant than those in the center. Those at the edges introduce discontinuities in the waveform and its derivatives which result in audible artifacts. As one possible solution, we include a weighted loss function which penalizes the prediction error of those samples that are found on the edges of a packet more than of those in the center. This is achieved by multiplying the absolute error for each sample by an inverse Hanning window:

$$w[n] = 1 - \sin^2\left(\frac{\pi n}{N-1}\right)$$

where $n = 0, 1, 2, ..., N-1$ is the sample index within a packet and $N = 128$ is the packet length. This approach is very similar to other nonlinear weighting techniques, for example, C-loss, Hinge Loss and other approaches used to handle data imbalances in machine learning [28]. Our assumption is that the network, by focusing on the

edge regions of each packet, will autonomously learn how to generate seamless transitions.

V. Performance Assessment

A. Database and Audio Acquisition Setup

For the purpose of this paper, all of the results are reported for a single instrument, violoncello. The recordings were made in a consistent manner with the same player and music and the same acoustical setup (microphone, positions and room). This was done to reduce the influence of these factors on our modeling results. A total of 15 hours of audio recording was collected. The cellist was practicing Bach's Six Suites for Violoncello over the course of a few weeks. These are extremely raw, unedited recordings including instrument tuning, scales practice and occasional improvisations. For the sake of the experiment, the database can be regarded as representative of a rehearsal of classical music such as might take place in a NMP session. The room was fairly reverberant with a high ceiling and sparsely furnished. The dataset is available online [29].

For the autoregressive approach, since it only requires storing of a set of past buffers, we do not create training data. For neural networks, we need to build a training and validation set to optimize the parameters.

Using a high-speed wired LAN with a gateway router coupled to WAN modem for either fiber-to-the-home, digital subscriber line or cable access to the Internet, packet dropouts usually happen only as a very small percentage of the total number of packets transmitted. Statistical distributions of packet loss probabilities are extremely variable, as they are highly depending on network traffic conditions. Therefore, for training purposes, we presume that packet dropouts are isolated and can happen at any instant. We sample time windows corresponding to the content carried by one packet at random points in the cello recordings. In particular, the missing packets thus selected amount to 1% of the available audio data. For each packet, we extract the respective spectro-temporal context and store the previous valid packet. A point to note is that a higher sampling would yield more training data and thus improved performance. This choice, however, was mainly due to the limited computational resources at our disposal.

The experiments use contexts of either 2 s or 8 s (storing intermediate representations of 2 s, 4 s and 8 s into 3 channels) that were (down-)sampled at 16 kHz (from 48 kHz) and a packet size of 128 samples. The goal of the experiments is: given the previous valid packets which have been received, can we predict the next packet?

Data input from the training and test sets were normalized to have a unity peak amplitude at each hop. For example, in the 2 s context case each new 2 s window-full would be normalized. By minimizing the L_1 norm between the predicted and the target waveforms, the model is biased towards reducing the errors of higher amplitude signals rather than lower amplitude signals. Context normalization avoids this amplitude-dependent performance bias.

Unfortunately, this also boosts packets which only have low-amplitude background noise as their content. However, since only a small fraction of packets in the database are noise-like, the neural network will mainly learn how to model deterministic signals, which in this case are mostly music signals, and will apply the learned model to noise-like signals, too.

B. Numerical Evaluation

We evaluate the latent space-based ML method described above and compare it to a baseline AR(p) model implemented using the *statsmodels* Python library [30]. Only isolated, single packet dropouts have been tested to date. Evaluation was performed on a held out test set. This test data consists of a single track of just over 11 minutes that was not present in the training dataset, for a total of 646 missing packets. As mentioned earlier, this test set vs. training set split does not affect AR(p) performance, as AR models estimate coefficients only from the previous history of the signal. An advantage of the AR model is that it can update its parameters from observing short-time correlations in the input stream, whereas ML methods derive a fixed set of parameters from *all* of the entire training set.

In comparing algorithm performance using context time scales of 2 s vs. 8 s, we noted that longer contexts do not improve the validation loss. For the comparison, we trained the network shown in Fig. 1 using two different kinds of spectro-temporal inputs. The first one consisted of the 2 s, 4 s and 8 s representation frames stacked in the first, second and third input channel of the convolutional neural network, respectively. The second, for compatibility with the same model architecture, consisted of the 2 s mel-spectrogram replicated it in all three input channels. The ML method provided equivalent results in both test cases. Therefore, the shorter 2 s context was used for all

further experiments. This finding indicates that PLC is not a long-term dependency problem, a fact reinforced also by the competence of the AR model in our testing scenario.

All of the results were evaluated on a Tesla K80 GPU using TensorFlow 1.15. In terms of achieving an eventual real-time system, things look favorable. The GPU-based system ran roughly 10 times faster than real time when computing the mel-spectrogram and doing a forward pass using the trained system. With a CPU-based system, the forward pass could not be computed in real time (defined as completing the process within the interval between the arrival of two packets). In our case, with a 1.8 GHz Intel Core i5 processor, it ran approximately 5 times slower than real time. Future improvements in both computational power and in the architectural simplification of deep learning-based models make it hopeful that such models can become tractable in real time without relying on hardware acceleration.

While perceptually-informed objective measures such as PEAQ [31] have been developed for assessing the quality of audio codecs, there is yet no consensus among the research community on whether these methods could prove reliable in evaluating the effectiveness of PLC algorithms [32], [33]. Hence, we measure the performance of both the proposed ML method and the AR baseline as the average absolute error between the predicted waveforms and actual ones. As described above, all tests involve normalized waveforms such that the peak is always one. This criterion avoids dependency on the amplitude of the signal. In the testing scenario described, we see that the ML-based approach outperforms the AR model within certain regimes.

The 128 sample packet size used in the experiment represents 8 ms of audio. The AR model comparison was run with 1024 and 2048 samples of history (64 ms and 128 ms, respectively) which were used for the conditional maximum likelihood estimation of the regression parameters via OLS. Its performance therefore depends on both the amount of history and the order p of the filter applied. AR(p) only achieves parity with the ML method with a 128 ms window and order above 57 as seen in Fig. 2. In our experiments, the AR model appears effective in predicting short gaps of missing audio (128 samples). However, AR(p) is able to outperform the ML method only with a fairly high number of regression parameters. If we were to test the PLC system on the longer gaps of bursty packet losses,

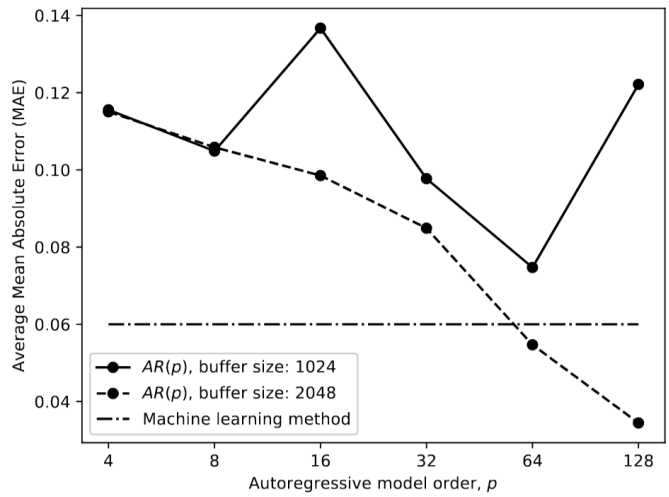

Fig. 2. Average Mean Absolute Error (MAE) of the autoregressive model AR(p) as a function of the number p of regression coefficients estimated using the last 1024 and 2048 valid samples (solid and dashed line, respectively) plotted against the average MAE of the proposed machine learning method (dash-dotted line).

then either the number of parameters to be estimated and thus the amount of history required would quickly become unmanageable or the results may converge to a zero signal after a few ms (a longer-term tendency for AR models). In this respect, ML-based approaches could provide an alternative solution.

VI. Conclusion

We described a neural network-based technique for packet loss concealment, specifically designed to be implemented in real-time audio streaming applications for networked music performance. The novel hybrid approach proposed in this paper marries two key elements. The first one is the spectral carry-forward of short-term (2 s) mel-spectrum content from a sufficiently large database of music recordings using latent vectors. The second one is a technique for the correct synthesis of a missing packet conditioned on a latent code. Used together, these methods predict and generate a packet which can be inserted in place of missing audio data. We believe that with future improvements to computational power, machine learning-based PLC will become pervasive for all types of audio signals. Investigations of possible improvements to the latent representation, as well as possible modifications of the synthesis block are envisioned, possibly by replacing the fully-connected neural network with a state-of-the-art recurrent module. Validating the experimental results via subjective listening tests in a real-world NMP scenario, generalizing

the approach to tackle a wide variety of audio signals (including different musical instruments, genres, players and recording environments) and extending it to prediction of longer excerpts (for example, those occurring in the case of bursty packet losses) will also be objectives of further studies.

ACKNOWLEDGEMENTS

We acknowledge the advice and encouragement of Andrea Bianco and thank Klaus Greff for the useful discussions.

REFERENCES

[1] Alexander Carôt and Christian Werner, "Fundamentals and principles of musical telepresence," *J. Sci. Technol. Arts*, vol. 1, no. 1, pp. 26–37, 2009.

[2] Cristina Rottondi, Michele Buccoli, Massimiliano Zanoni, Dario Garao, Giacomo Verticale, and Augusto Sarti, "Feature-based analysis of the effects of packet delay on networked musical interactions," *J. Audio Eng. Soc.*, vol. 63, no. 11, pp. 864–875, 2015.

[3] Chris Chafe, "I am streaming in a room," *Frontiers in Digital Humanities*, vol. 5, pp. 27, 2018.

[4] Juan-Pablo Cáceres and Chris Chafe, "Jacktrip: Under the hood of an engine for network audio," *J. New Music Res.*, vol. 39, no. 3, pp. 183–187, 2010.

[5] Alexander Carôt and Christian Werner, "Distributed network music workshop with Soundjack," in *Proc. of the 25th Tonmeistertagung*, 2008.

[6] Carlo Drioli, Claudio Allocchio, and Nicola Buso, "Networked performances and natural interaction via LOLA: Low latency high quality A/V streaming system," in *Proc. of the Int. Conf. on Information Technologies for Performing Arts, Media Access, and Entertainment*, 2013, pp. 240–250.

[7] Petr Holub, Jiří Matela, Martin Pulec, and Martin Šrom, "UltraGrid: Low-latency high-quality video transmissions on commodity hardware," in *Proc. of the 20th ACM Int. Conf. on Multimedia*, 2012, pp. 1457–1460.

[8] Cristina Rottondi, Chris Chafe, Claudio Allocchio, and Augusto Sarti, "An overview on networked music performance technologies," *IEEE Access*, vol. 4, pp. 8823–8843, 2016.

[9] Ahmed Badr, Ashish Khisti, Wai-Tian Tan, and John Apostolopoulos, "Perfecting protection for interactive multimedia: A survey of forward error correction for low-delay interactive applications," *IEEE Signal Processing Magazine*, vol. 34, no. 2, pp. 95–113, 2017.

[10] Bong-Ki Lee and Joon-Hyuk Chang, "Packet loss concealment based on deep neural networks for digital speech transmission," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 24, no. 2, pp. 378–387, 2015.

[11] Reza Lotfidereshgi and Philippe Gournay, "Speech prediction using an adaptive recurrent neural network with application to packet loss concealment," in *Proc. of the 2018 IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2018, pp. 5394–5398.

[12] John Lazzaro and John Wawrzynek, "A case for network musical performance," in *Proc. of the 11th Int. Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2001, pp. 157–166.

[13] Jussi Virolainen and Pauli Laine, "Methods and apparatus for transmitting MIDI data over a lossy communications channel," May 24 2005, US Patent 6,898,729.

[14] Jason Robert Nelson, Allen James Heidorn, and Robert John Cox, "Reliable real-time transmission of musical sound control data over wireless networks," Mar. 21 2017, US Patent 9,601,097.

[15] Ye Wang, "System and method for error concealment in digital audio transmission," Nov. 4 2008, US Patent 7,447,639.

[16] Marco Fink and Udo Zölzer, "Low-delay error concealment with low computational overhead for audio over IP applications.," in *Proc. of the 17th Int. Conf. on Digital Audio Effects*, 2014, pp. 309–316.

[17] Henning Sanneck, Alexander Stenger, K Ben Younes, and Bernd Girod, "A new technique for audio packet loss concealment," in *Proc. of the 1996 IEEE Global Telecommunications Conference*, 1996, pp. 48–52.

[18] Guoqiang Zhang and W Bastiaan Kleijn, "Autoregressive model-based speech packet-loss concealment," in *Proc. of the 2008 IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2008, pp. 4797–4800.

[19] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu, "Pixel recurrent neural networks," *arXiv preprint arXiv:1601.06759*, 2016.

[20] Albert Haque, Michelle Guo, Prateek Verma, and Li Fei-Fei, "Audio-linguistic embeddings for spoken sentences," in *Proc. of the 2019 IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2019, pp. 7355–7359.

[21] Albert Haque, Michelle Guo, and Prateek Verma, "Conditional end-to-end audio transforms," *arXiv preprint arXiv:1804.00047*, 2018.

[22] Prateek Verma and Julius O. Smith, "Neural style transfer for audio spectrograms," *arXiv preprint arXiv:1801.01589*, 2018.

[23] Aaron van den Oord, Oriol Vinyals, et al., "Neural discrete representation learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 6306–6315.

[24] Kazuhiro Kondo and Kiyoshi Nakagawa, "A speech packet loss concealment method using linear prediction," *IEICE Trans. on Information and Systems*, vol. 89, no. 2, pp. 806–813, 2006.

[25] Colin Perkins, Orion Hodson, and Vicky Hardman, "A survey of packet loss recovery techniques for streaming audio," *IEEE network*, vol. 12, no. 5, pp. 40–48, 1998.

[26] Sander Dieleman, Aaron van den Oord, and Karen Simonyan, "The challenge of realistic music generation: Modelling raw audio at scale," in *Advances in Neural Information Processing Systems*, 2018, pp. 7989–7999.

[27] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[28] Yichao Wu and Yufeng Liu, "Robust truncated hinge loss support vector machines," *J. Am. Stat. Assoc.*, vol. 102, no. 479, pp. 974–983, 2007.

[29] Chris Chafe, "Bach cello suites data repository," Web: https://ccrma.stanford.edu/~cc/som/bachCello (accessed July 1, 2020).

[30] Skipper Seabold and Josef Perktold, "Statsmodels: Econometric and statistical modeling with Python," in *Proc. of the 9th Python in Science Conference*, 2010.

[31] Thilo Thiede et al., "PEAQ — the ITU standard for objective measurement of perceived audio quality," *J. Audio Eng. Soc.*, vol. 48, no. 1/2, pp. 3–29, 2000.

[32] Marco Fink, Martin Holters, and Udo Zölzer, "Comparison of various predictors for audio extrapolation," in *Proc. of the 16th Int. Conf. on Digital Audio Effects*, 2013, pp. 1–7.

[33] Ala Khalifeh, Abdel-Karim Al-Tamimi, and Khalid Darabkh, "Perceptual evaluation of audio quality under lossy networks," in *Proc. of the Int. Conf. on Wireless Communications, Signal Processing and Networking*, 2017, pp. 939–943.