Audio Engineering Society

Convention Express Paper 112

Presented at the 155th Convention
2023 October 25–27, New York, USA

Application of ML-Based Time Series Forecasting to Audio Dynamic Range Compression

Pascal Brunet[1], Yuan Li[1], and Soohyun Kim[2]

[1]*DMS-Audio Lab, Samsung Research America*
[2]*CCRMA, Stanford University*

Correspondence should be addressed to Pascal Brunet (p.brunet1@samsung.com)

ABSTRACT

Time Series Forecasting (TSF) is used in astronomy, geology, weather forecasting, and finance to name a few. Recent research [1] has shown that, combined with Machine Learning (ML) techniques, TSF can be applied successfully for short-term predictions of music signals. We present here an application of this approach for predicting audio level changes of music and appropriate Dynamic Range Compression (DRC). This ML-based look ahead prediction of audio level allows to apply compression just-in-time, avoiding latency and attack/release time constants, which are proper to traditional DRC and challenging to tune.

1 Introduction

Compressors are typically used to raise levels of quiet sounds and attenuate loud ones [2]. They are commonly used in production and broadcast of audio content. Main applications are found in TV, radio, automotive to work around high background noise and accommodate limited dynamic playback, in music production to shape transients and increase the perceived loudness. Commercial advertisements use DRC to blast their message.

Traditional DRCs are generally misunderstood and challenging to tune [3, 4]. Their parameters affect the sound quality. For example, attack and release time constants are often program dependent. Some common problems found in state-of-art compressors are: latency due to look-ahead delay and attack/release time constants, tuning complexity (especially for multi-band compressors), distortion, breathing, pumping, washed out, no punch sound, stereo image shift. And specific to multi-band compressors, they are: intermodulation between frequency bands, changing timber and tonal balance, resonances, phase shifting...

To alleviate the tuning complexity, current research on DRC use ML techniques to automatically tune or emulate traditional systems [5, 6, 7, 8, 9]. No research has been done so far (to our knowledge)to rethink the DRC design fundamentally. This paper presents a new innovative approach. Our invention [Patent pending] use Time Series Forecasting (TSF) to predict the levels on an audio stream in real-time and compress/limit its dynamic range. The TSF is based on ML techniques.

Time Series Forecasting (TSF) is used in astronomy, geology, weather forecasting, and finance to name a few [10, 11, 12, 13]. Recent research [1] has shown that, combined with Machine Learning (ML) techniques, TSF can be applied successfully for short-term predictions of music signals. We present here an application of this approach for predicting audio level changes of music and performing appropriate DRC. This ML-based look ahead prediction of audio level allows to apply compression just-in-time, avoiding latency and attack/release time constants, which are proper to traditional DRC and challenging to adjust appropriately.

In the following sections, we will first present the principles of TSF with machine learning. Then we will show how TSF can be used in music forecasting and applied to DRC. We will present the outline of our ML-DRC algorithm. Next we will describe the ML training and tests, the performances metrics and compare different networks architectures. We will compare ML-DRC with traditional DRC, both in terms of objective results, and subjective results with a listening test. We will finish by discussing the advantages and limitations of our approach, propose future lines of work and conclude.

2 Principles

2.1 Time Series Forecasting (TSF)

Time series data is a sequence of observations or measurements collected at successive points, where each data point corresponds to a specific time period. The order of the sequence is significant, as it holds information about trends, seasonality, and patterns.

TSF is a specific application within time series analysis that predicts future values based on previous observations. One conventional approach is Auto-regressive Integrated Moving Average (ARIMA): ARIMA captures auto-correlation, differences, and moving average components to forecast future values. They are effective for stationary time series data. Another approach is Exponential Smoothing: it captures trends and seasonality. With the rise of big data and enhanced computational power, ML has emerged as an essential component of TSF models in recent years. Popular ML approaches for TSF include Recurrent Neural Network (RNN), Long Short-term Memory (LSTM), Gated Recurrent Units (GRUs), Attention-based Transformers, etc ... [14, 15]

By utilizing past information, ML models can learn the relationship between input characteristics and upcoming values. The resulting model can forecast the values at future time instances. Given a time series $x_1, x_2, ...x_t$, where x_t is a vector of N input features observed or measured at time t, the goal is to design a model to predict $\hat{y}_{(t+1)}$ at a future time $t+1$ [14]. A functional relationship learned by the ML models of one-step ahead forecasting is as follows:

$$\hat{y}_{(t+1)} = f(x_{t-k}, ..., x_{t-1}, y_{t-k}...y_{t-1}) \quad (1)$$

where $y_{t-k}, ..y_{t-1}$ are the target values observed from time $t-k$ to $t-1$; $x_{t-k}, ...x_{t-1}$ are the vector of observed input features from time $t-k$ to $t-1$; f is the prediction function learned the models; k is the depth of past time steps.

2.2 Dynamic Range Compression (DRC)

DRC is a signal processing technique used in audio and music production to control the difference between the loudest and quietest parts of an audio signal. It involves reducing the dynamic range of an audio signal by attenuating or compressing the amplitude of the signal's peaks, making the overall audio signal more balanced and consistent in volume. This technique is commonly applied to improve audio recordings' perceived loudness and clarity, ensuring that quieter sounds are still audible without distorting or clipping the louder parts.

Here are the main parameters of a DRC compressor. *Threshold Setting*: A threshold is a certain level below which the signal is considered quiet or insignificant. When the audio signal surpasses this threshold, dynamic range compression starts to take actions. *Compression Ratio*: The compression ratio determines how much the audio signal above the threshold will be reduced in amplitude. *Attack Time*: This parameter determines how quickly the compressor responds once the audio signal crosses the threshold. A shorter attack time means the compressor reacts swiftly to changes in volume. *Release Time*: Release time governs how long it takes for the compressor to return to normal operation after the input signal falls below the threshold again. *Knee*: It defines the gradualness with which compression is applied as the signal approaches the threshold. A "soft knee" introduces compression gradually before the threshold is fully crossed, resulting in a smoother transition. *Make-up Gain*: After compression, the overall signal level might be reduced. Make-up gain

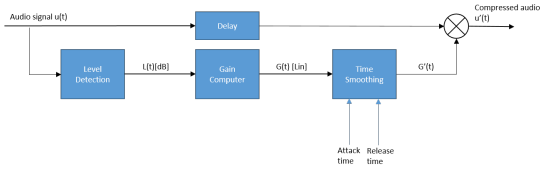

Fig. 1: Traditional DRC

compensates for this reduction by amplifying the compressed signal to achieve a desired output level. *Delay*: A delay is often inserted in the direct path in digital implementations to compensate for the attack time and other lags in the side channel.

Knowing the incoming audio level allows the compressor to anticipate changes in the audio signal before they occur, helping to provide more accurate and refined compression. It helps the compressor respond more accurately to rapid changes in the audio signal. While a look-ahead level prediction allows the compressor to apply compression just in time, it avoids latency and attack/release time constants tuning, which are proper to traditional DRC and challenging to adjust appropriately.

3 Design

Based on TSF principles, our algorithm uses look-ahead level prediction to adjust the levels on an audio stream in real-time and compress its dynamic range. The level prediction uses ML techniques. The main advantages of this approach are: no latency, simpler tuning, and better sound quality. Notably, the attack and release time constants are eliminated which are difficult to tune and impact the sound quality, eliminating potential breathing, pumping effects.

A block diagram is shown in Fig. 2, giving an overview of the full algorithm for streaming applications. The pseudo-code is given in Algorithm 1.

Fig. 3 shows a typical example of compression law where input levels are limited and low levels are raised.

4 Network Architectures & Training

4.1 Hyperparameters

For time-series forecasting, we considered single-step time-series forecasting with a memory depth M of 5,

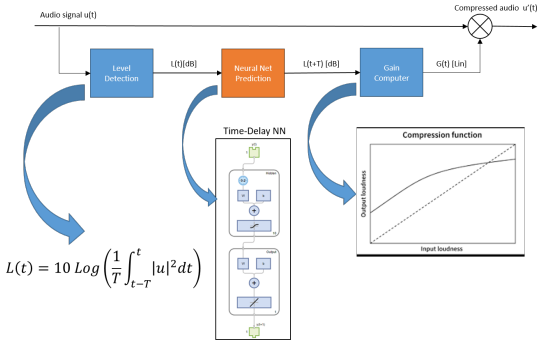

Fig. 2: Overview of ML-DRC

Algorithm 1 DRC algorithm with ML-based TSF

$f_s \leftarrow$ sampling rate [Hz]
Leq duration [in s]: $T \leftarrow 1$
Frame size [in samples]: $N \leftarrow [T f_s]$
Initial net state: $L_{state} \leftarrow$ vector of initial Leq's
Initial gain (linear value): $G \leftarrow 1$
while not end of input audio stream **do**
  Read frame **x** from input stream
  Calculate current Leq: $L \leftarrow 10\log\left(\frac{1}{N}\sum_{n=1}^{N}|x_n|^2\right)$
  Apply gain to **x**: $\mathbf{y} \leftarrow G\,\mathbf{x}$
  Write frame y to output stream
  Leq prediction and state update with Neural Net: $[L_{next}, L_{state}] \leftarrow Net(L, L_{state})$
  Apply compression law: $Lr \leftarrow f(Lnext)$
  Compute next gain (linear value): $G = 10^{(Lr-Lnext)/20}$
end while

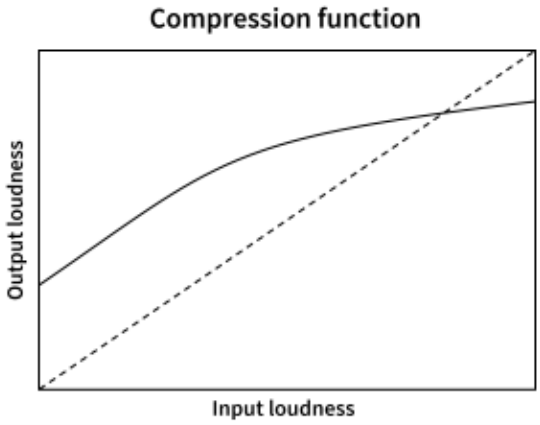

Fig. 3: Example of Compression Law

which means a neural network predicts the audio level of one future time frame based on five preceding time frames, including the current time frame. An input for a neural network is an array of five sequential audio level values, and the output is one audio level prediction value. We have tested with M from 1 to 15, the precision increased as increasing M. However, in the range of 5 to 15, the improvement in MAE (Mean Absolute Error) loss was only different at the second decimal point in the dB scale, which, therefore, is not worth increasing computation.

The size of hidden nodes was set to 32 based on the input memory depth to ensure that the network is capable to represent our problem without over-fitting.

Each time frame is one second in length with no overlap, and the audio level value of each time frame is the rms level of the audio signal for the corresponding one-second window. We have also tested with different time frame length T of 0.1, 0.2, 0.5, 1 and 2 seconds, and the training loss decreased as the time frame length increased, which means better prediction accuracy. However, it also means that the time series of rms levels has become more predictable with less time resolution because averaging with a longer time length evens out the detailed changes in the audio level dynamics. With less time resolution, we also lose the resolution of the dynamic range control, which is the ultimate goal of our Algorithm 1. We chose 1 second as a trade-off between prediction accuracy and time resolution since it preserves enough time resolution for audio level fluctuations.

4.2 Training Setup

For neural network models, we conducted experiments on FCN (Fully-Connected Network), LSTM (Long Short-Term Memory), GRU (Gated Recurrent Units), and Transformer Encoder. We then compared their performances on the same training and test dataset. They all have the same number of hidden state features, 32, for the same input size, 5. The output size of FCN and the Transformer Encoder was set to 1. While the LSTM and GRU output an array of length 5, we only took the last element of the array to predict the subsequent time frame. We used MAE loss and Adam optimizer with a learning rate of 0.0001 and a batch size of 1 with random shuffling. During training process, we kept the optimal performance for each neural network architecture when both the training set and test set hit the minimum loss value for parallel comparison.

We chose MAE loss over MSE loss as MAE loss is less sensitive to outliers in a dataset. During the training with our training set, there were exceptionally significant prediction errors, which are statistical outliers, and occurred only in a handful of moments, for instance, when there was a silent moment in the training data audio. We tested both MSE and MAE loss, and the training was more stable with MAE loss as MAE loss was less sensitive, even if the outlier error occurred over a silent period.

As shown in table. 1, we tested our neural networks for both the linear scale dataset, in which audio level values are between 0 and 1, and the dB scale dataset, in which audio level values are below 0 dB and can go down around -40 dB.

The baseline method for our experiments was simply using the previous time frame's audio level as the prediction for the current time frame's audio level. This represents the real-time application of a traditional dynamic range compressor where processing on a current time frame has to rely on the previous time frame's audio level. Otherwise, there should be latency longer than one time frame to wait until receiving the current time frame's data.

Network		Training set	Test set
Baseline		1.19	2.26
FCN	Linear	1.16	2.69
	dB	1.13	2.05
LSTM	Linear	1.14	1.87
	dB	1.13	1.87
GRU	Linear	1.13	1.88
	dB	1.13	1.92
Transformer Encoder	Linear	1.17	1.98
	dB	1.18	2.00

Table 1: MAE error (dB) comparison for different neural network models.

4.3 Training Results

Our training results for different neural network models are shown in Table 1. LSTM, especially with the dB scale, showed the best performance in the audio level change prediction both for the training and test set. However, the performance differences among different neural network models were marginal. One exceptional case was FCN with the linear scale, where the test set

error was bigger than the baseline's. Since the dB scale is logarithmic, the same error in the linear scale is not the same when converted into the dB scale. Even if there is a small under-prediction error in the linear scale, the error in the dB scale explodes or becomes undefined if the prediction level value goes below 0 when the actual audio level is very low and close to 0 in the linear scale. For such cases, FCN with the linear scale under-predicted several times, and it induced huge errors in the dB scale. Otherwise, every neural network model showed better performance than the baseline both for the training and test set.

4.4 Prediction Error Analysis

Since we are interested in the dynamic range of the output audio level for whether we get sufficient dynamic range compression, we have to investigate not only the prediction accuracy but also the prediction variance for how the variance of prediction error can affect the dynamic range compression.

Let us consider a compression law $f(L)$, where L is a raw audio level, $f(L)$ is the compressed audio level, and the compression ratio r is given as $\frac{1}{r} = \frac{\partial f(L)}{\partial L}$. Due to the prediction error δ of a neural network, we apply this compression law on the predicted audio level $\widehat{L} = L + \delta$, so the compression gain we get from our algorithm [Algo. 1] is $f(\widehat{L}) - \widehat{L}$. However, we then apply this compression gain on the raw audio level L, which results in the final output audio level $\widetilde{L}$ given as

$$\begin{aligned} \widetilde{L} &= L + f(\widehat{L}) - \widehat{L} = f(L+\delta) - \delta \\ &\simeq f(L) + \left(\frac{\partial f}{\partial L}\right)\delta - \delta = f(L) - \left(1 - \frac{1}{r}\right)\delta, \end{aligned} \tag{2}$$

where the equality precisely holds when f is linear.

We are interested in the dynamic range of the output audio level $\widetilde{L}$, which can be quantified as the variance $\mathrm{Var}(\widetilde{L})$ based on Eq.(2) as follows

$$\begin{aligned} \mathrm{Var}(\widetilde{L}) = (1/r)^2\mathrm{Var}(L) + (1-1/r)^2\mathrm{Var}(\delta) \\ - 2(1/r)(1-1/r)\mathrm{Cov}(L,\delta) \end{aligned} \tag{3}$$

for $f(L) \sim \frac{L}{r}$. Eq.(3) indicates the prediction error δ brings an additional variance and covariance term into $\mathrm{Var}(\widetilde{L})$, compared to the ground truth compression result, $\mathrm{Var}(f(L)) = (1/r)^2\mathrm{Var}(L)$, which is a performance loss in terms of dynamic range compression. The weighting coefficient of $\mathrm{Var}(\delta)$ and $\mathrm{Cov}(L,\delta)$ are already on par with that of $\mathrm{Var}(L)$ for $r = 2$, and even get more dominant as the compression ratio r increases.

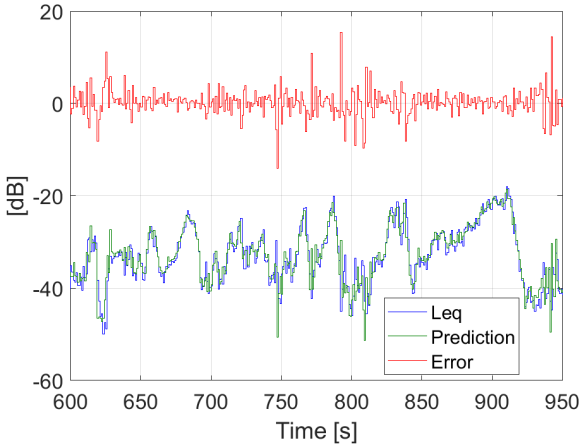

Fig. 4: Test prediction- Classic orchestral music (extract)

Therefore, if the variance of a raw audio level is on par with the variance of a neural network's prediction error, we do not get enough compression with our algorithm.1. For instance, as shown in Table. 2, a pop music piece typically has a narrow dynamic range and thus a small $\mathrm{Var}(L)$, which is on par with $\mathrm{Var}(\delta)$ and $\mathrm{Cov}(L,\delta)$. In such a case, we only get the actual compression ratio of 1.16 even though we have set the compression ratio $r = 2$ for our algorithm. In contrast, for a classical music piece with a large dynamic range, in which $\mathrm{Var}(L)$ is multiple times bigger than $\mathrm{Var}(\delta)$ and $\mathrm{Cov}(L,\delta)$, we get the actual compression ratio of 1.62, much closer to the compression ratio $r = 2$ we set. This demonstrates that our algorithm.1 is more effective for the audio material which has a large dynamic range.

5 Results & Discussion

LSTM gives the best results in term of forecasting precision with an MAE of 1.87 dB (cf Table 1).
Fig. 4 shows an example of prediction results on test material. Nonetheless the limited forecasting precision restrict ML-DRC to the compression of large dynamic range material (e.g. classical music). Pieces with a small dynamic range, similar to the precision, hardly get compressed (cf table 2).

Silent parts are an issue (cf fig. 5) and will need special processing. Start of song exhibit issues as well and will

Genre of audio material	Var(L) (raw)	Var(δ)	Cov(L, δ)	Var($\widetilde{L}$) (compressed)	Actual compression ratio
Pop	7.1	6.1	-4.0	5.3	1.16
Classical	33	8.3	-4.6	12.6	1.62

Table 2: Prediction error δ's effect on dynamic range compression. (FCN trained in dB scale, Compression ratio $r = 2$)

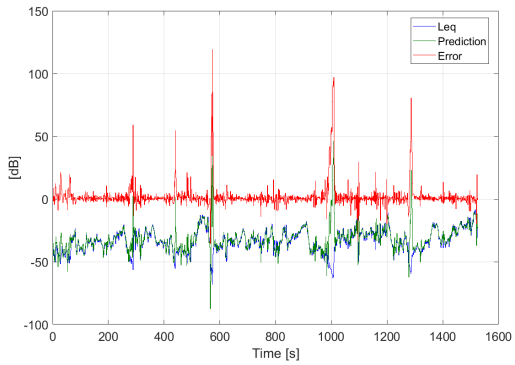

Fig. 5: Test prediction- Classic orchestral music (entire work- 3 movements)

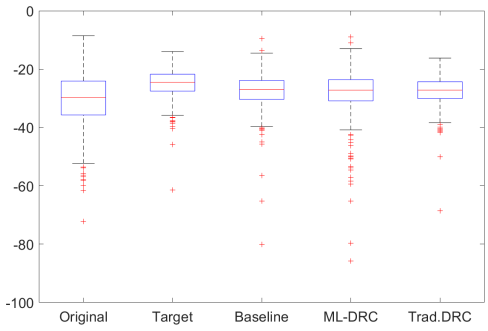

Fig. 7: Level statistics on compressed music. Target: offline compression

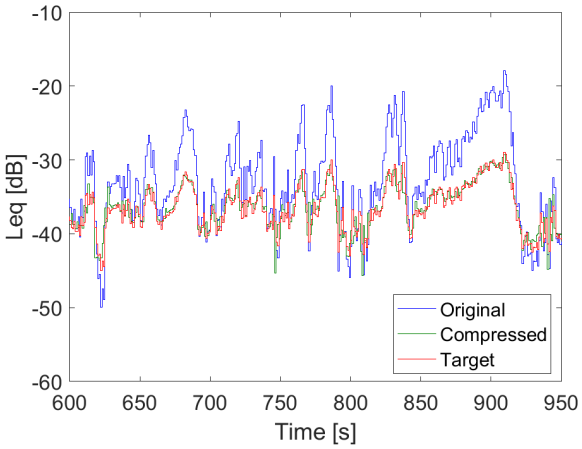

Fig. 6: Example of compression obtained with ML-DRC. Target: offline compression

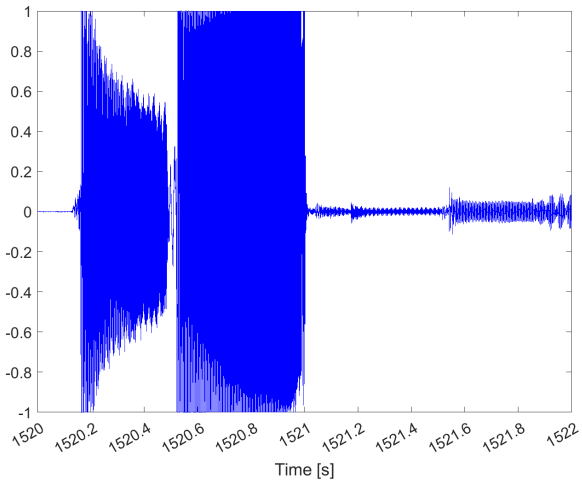

Fig. 8: Example of compression error obtained with baseline method

require some initialization process. We leave that for future development.

Fig. 6 shows an example of compression achieved on the extract of music shown in fig. 4.

For comparison purpose, the same piece of music has been compressed using different methods: offline compression, baseline, ML-DRC, and traditional DRC. Offline compression yields perfect results and is denoted target. For traditional DRC, we used a Matlab compressor, part of the Audio ToolBox with the following parameters:

- Threshold : -120 dB,
- Ratio : 2,
- KneeWidth : 0 dB,
- AttackTime : 50 ms,
- ReleaseTime : 200 ms,
- MakeUpGain : 49.7 dB.

The level statistics of the compression results along with the original (uncompressed) are shown as box plots in fig.7. We see that the interquartile range (box height) fot the levels of the compressed files are all close to the target. ML-DRC exhibits the biggest variability (whiskers) and the most negative outliers (red crosses).

Baseline compression method is attractive because of its efficiency and simplicity but it can create strong audible errors. In some cases it creates high positive gains that produces very loud frames (cf fig. 8). This happens after a momentary silence in the music. This problem didn't happen with ML-DRC that tends on the contrary to create negative outliers (too quiet frames) (cf fig. 7). In general,we have observed that ML-DRC tend to produces less outliers than baseline method, because forecasting is based on several past levels, instead of only one. This tends to smooth the predicted values.

An attempt has been made to run blind listening tests with experienced listeners without clear results. Subjective opinions on the different methods were collected in an informal listening test involving seven participants. Four twelve-second extracts were used to compare ML-DRC, Baseline DRC, traditional DRC (Matlab function) with the following content: orchestral music, horse riding scene with background music and narration voice, in-car dialog, guitar tune. The compression method was unknown from the participants and the order of presentation was random. Assessors generally found it difficult to discriminate between the listening conditions, and were unable to elicit a preference. The compressed files obtained with traditional DRC, ML-DRC and even with baseline DRC could not be distinguished.

From this first study, we can outline future directions of work: design methods to manage start & stop of material, and silent sections, improve forecasting precision and robustness, ensure good generalization by training of large data set. The principle of ML-DRC could be extended to limiter applications by using short frame duration (e.g. 1 ms) ...

6 Conclusion

Based on this study, we can conclude that DRC based on TSF with ML techniques is a viable approach for dynamic range compression of large dynamic range material, which is the principal aim of DRC. In particular, blind listening tests have shown that the sound quality of the compressed of material is on par with traditional DRC. Some work needs to be pursued to guarantee robustness, especially with processing of beginning & end of pieces and silences, and generally improve the precision. The simplicity of use, with no tuning required, would be a plus for the sound engineer, or even for the end user.

7 Acknowledgments

Samsung Electronics and Samsung Research America supported this work. The authors would like to thank the entire staff of Samsung's US Audio Lab who helped with all aspects of this research, offered insightful suggestions, and contributed to this work.

References

[1] Gentile, I., "Music Audio Signal Prediction using Machine Learning," 2022.

[2] Giannoulis, D., Massberg, M., and Reiss, J. D., "Digital dynamic range compressor design—A tutorial and analysis," *Journal of the Audio Engineering Society*, 60(6), pp. 399–408, 2012.

[3] Izhaki, R., *Mixing audio: concepts, practices, and tools*, Routledge, 2017.

[4] Zölzer, U., *Digital audio signal processing*, John Wiley & Sons, 2022.

[5] Sheng, D. and Fazekas, G., "A feature learning siamese model for intelligent control of the dynamic range compressor," in *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2019.

[6] Mimilakis, S. I., Drossos, K., Virtanen, T., and Schuller, G., "Deep neural networks for dynamic range compression in mastering applications," in *Audio Engineering Society Convention 140*, Audio Engineering Society, 2016.

[7] Hawley, S. H., Colburn, B., and Mimilakis, S. I., "SignalTrain: Profiling audio compressors with deep neural networks," *arXiv preprint arXiv:1905.11928*, 2019.

[8] Steinmetz, C. J. and Reiss, J. D., "Efficient neural networks for real-time modeling of analog dynamic range compression," *arXiv preprint arXiv:2102.06200*, 2021.

[9] Wright, A., Välimäki, V., et al., "Grey-box modelling of dynamic range compression," in *Proc. Int. Conf. Digital Audio Effects (DAFx)*, pp. 304–311, 2022.

[10] Chatfield, C., "Time-series forecasting," *Significance*, 2(3), pp. 131–133, 2005.

[11] Sezer, O. B., Gudelek, M. U., and Ozbayoglu, A. M., "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019," *Applied soft computing*, 90, p. 106181, 2020.

[12] Chimmula, V. K. R. and Zhang, L., "Time series forecasting of COVID-19 transmission in Canada using LSTM networks," *Chaos, solitons & fractals*, 135, p. 109864, 2020.

[13] Deb, C., Zhang, F., Yang, J., Lee, S. E., and Shah, K. W., "A review on time series forecasting techniques for building energy consumption," *Renewable and Sustainable Energy Reviews*, 74, pp. 902–924, 2017.

[14] Bryan, L. and Stefan, Z., "Time-series forecasting with deep learning: a survey," *Philosophical Transactions of the Royal Society A 2020*, 379, 2020.

[15] Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., and Sun, L., "Transformers in time series: A survey," *arXiv preprint arXiv:2202.07125*, 2022.