# Intelligent control method for the dynamic range compressor: A user study*

**Shubhr Singh, Gary Bromham, Di Sheng, György Fazekas**

*Centre for Digital Music (C4DM)*

*Queen Mary University of London*
*London, UK*

Music producers and casual users often seek to replicate dynamic range compression used in a particular recording or production context for their own track. However, not knowing the parameter settings used to produce the audio using the effect may become an impediment, especially for beginners or untrained users who may lack critical listening skills. We address this issue by presenting an automatic compressor plugin relying on a neural network to extract relevant features from a reference signal and estimate compression parameters. The plugin automatically adjusts its parameters to match the input signal with a reference audio recording as closely as possible. Quantitative and qualitative usability evaluation of the plugin was conducted with amateur, pro-amateur and professional music producers. The results established acceptance of the core idea behind the proposed control method across these user groups.

## 1 Introduction

Dynamic range compression (DRC) is the process of mapping the dynamic range of an audio signal to a smaller range, this may be implemented to reduce the signal level of higher peaks while leaving the quieter parts unchanged [1]. Dynamic range compression has the capability to make a sound punchier, richer and more impactful. However, if used without sufficient knowledge or acknowledgement of style or context [2], compressors can also suppress the musical dynamics and destroy the natural texture of the sound.

The most common set of parameters that are used to characterize DRC are *Threshold, Ratio, Knee, Attack* and *Release.*

- *Threshold* is the level above which compression starts
- Ratio is the amount of compression that will be applied to the signal
- *knee* controls whether the transfer characteristics of a compressor has a sharp (hard knee) or smooth (soft knee) transition around the threshold [1].
- *Attack* and *release* times determine how fast the compressor acts.

Controlling an audio effect like the DRC involves in-depth understanding of how each parameter affects the dynamics of the sound. This understanding often necessitate knowledge of signal processing too. DRC is a non-linear audio effect, hence the impact of each control parameter is not always perceptually obvious. Furthermore, since the controls may display a high degree of correlation to each other [3], low-level signal processing knowledge becomes imperative to achieve the desired output from the audio effect. Combining substantial practical experience with good insight into what happens at the signal level acts as a barrier for amateur users. They might be accustomed to describing the kind of sound they intend to produce through a semantic descriptor or a reference audio recording rather than through low-level signal processing terms. Even for professional music producers who are well versed with the requisite low-level knowledge, getting the right set of controls to produce the desired output is often a time consuming process. Consequently, intelligent music production tools which automate the parameter settings of a compressor are beneficial for both amateur and professional users.

Research into parameter automation of dynamic range compressors is not new. A significant body of previous work has focused on automating the parameters based on modeling the statistical properties of the audio signal before and after compression [4] or using side chain feature extraction from the input signal [1, 3]. While these approaches have shown promising results, they do not provide a way to map the low-level signal processing parameters to high-level concepts such as a semantic descriptor [5] or an example sound [6, 7].

*Correspondence should be addressed to: s.shubhr@qmul.ac.uk

Deep neural networks (DNNs) provide a suitable framework for the aforementioned task. They have been increasingly commonly used for learning a non-linear mappings between low-level features to high-level ones in both the audio and computer vision domains. They demonstrated promising results in a diverse range of audio signal processing tasks such as classification, de-reverberation [8, 9], mixing [10] and synthesis [11]. Recently, DNNs have been used to model DRC [12] where an autoencoder is used to map an un-processed audio to the processed audio and is conditioned on the vector of DRC controls.

We adopt the methodology introduced in [7], where a parameter automation approach for dynamic range compressor is proposed using a siamese neural network [13] and a reference audio signal. Since it is not always possible to verbalise complex audio engineering concepts, users might express what they want to hear through a reference audio example. For instance, an artist may refer to an excerpt from a track they already know. An audio engineer may also recall a previously produced track from their own library. The proposed methodology uses a reference audio example to estimate the DRC parameters, by attempting to bring the input signal closer to the designated reference in terms of dynamic range. This work integrates the deep neural network model proposed in [7] with a VST plugin[1] and evaluate its usability and acceptance in different use case scenarios across amateur, professional-amateur (Pro-am [14]) and professional users.

The rest of the paper is organised as follows. Section 2 outlines related work. Section 3 discusses the workflow of the auto compressor plugin. The user study is presented in Section 4, followed by outlining the results in Section 5. Critical analysis of the results is presented in Section 6 followed by conclusions and future work in Section 7.

## 2 Related work

In this section, we will briefly discuss existing research in the developing field of intelligent music production with a special emphasis on DRC. In [15], approaches to intelligent audio mixing have been divided into categories associated with three different methodologies.

- Grounded theory - This approach employs psychoacoustics and perceptual evaluation studies to understand the mixing process and subsequently the intention of the mixing engineer. It is resource intensive and not always reliable since mix engineers do not always follow a set of rules, making it difficult to encompass all variations within a particular framework, while decisions are often influenced by context [2].

- Knowledge engineering - This approach aims to integrate established knowledge/best practices into rules and constraints under which the system operates.

- Machine learning - This approach deals with mapping audio features to a particular output. The output is usually in line with fulfilling a particular aim such as automation of a single audio effect [16] or finding an optimal dynamic range of each instrument in a mix [3] or automating the parameters of a particular audio effect [1].

DNNs fall in the third category as the most recent development in context of intelligent music production. DNNs have been used for black box modelling of audio effects [17]. In [18], the Wavenet architecture was used to model audio distortion circuits, whereas in [19] recurrent networks was used to model a vacuum tube guitar amplifier. A popular variant of recurrent neural networks known as long short time memory networks was used in [20] to model a tube amplifier. Recently, temporal convolution networks have also been used to model audio effects [21, 22]. In [23], DNNs have been used to extract vocals from a mix. In [24], pretrained autoencoders have been used to map low-level signal processing features to dynamic range compression factors in such a way that it simulates the aesthetics of dynamic range processing. In [12], a network architecture has been proposed which intends to learn the process of dynamic range compression by synthesizing audio given the input-output pair parameters. The model used in our work is significantly different from all the above mentioned approaches as it intends to learn the low-level signal processing parameters associated with controlling the DRC from a reference audio signal. Our methodology is somewhat similar to the compressor proposed in [25], where the dynamic characteristics of an audio file is matched to some desired characteristics using a novel measure called *dynamic spread*.

A broad range of approaches have been proposed to automate the application of dynamic range compressors. For instance in [26], audio analysis and pattern recognition is used to enable recalling DRC parameters in similar musical contexts. The authors in [27] propose a system to configure audio effects, including the DRC, using semantic descriptors associated with crowed-sourced parameters, and show there is consistency across users in the use of semantic terms and intended modifications to the signal. An approach to reduce the number of user configurable parameters of the DRC is presented in [28]. Several researchers aim to automate the DRC in the context of automatic mixing. While these are less relevant in the context of the approach proposed in this paper, the reader is referred to a exhaustive review provided in [29].

## 3 Methodology

The following subsections outline the development of the compressor plugin that employs the proposed control mechanism. The plugin was developed with the aim of assessing the utility of the proposed automation approach in the context of different users and audio production workflows. The plugin is built using the VST standard applicable in a broad range of Digital Audio Workstations.

[1] `https://www.steinberg.net/en/company/technologies/vst3.html`

### 3.1 Overview

The Graphical interface (GUI) of the plugin is shown in Figure 1a. The plugin interface consists of controls for four key parameters of a compressor - ratio, threshold, attack and release. The compressor algorithm has been adopted from the open source SAFE project [5] to facilitate reproducibility of our study. The reference audio example can be dragged and dropped to the *list box* widget as shown in Figure 1.

The internal structure and data flow of the compressor plugin is shown in Figure 2. The plugin has been developed using the JUCE platform. It utilises a workflow that requires "learning" parameters from the reference audio and making a comparison with the audio to be processed. This type of workflow is common in the context of noise-reduction algorithms that rely on a noise fingerprint. A general use case scenario of the plugin and its typical workflows can be outlined as follows:

1. The plugin has two tab widgets providing views for two alternative workflows. The first tab exhibits all conventional controls of a dynamic compressor, while the second tab has a single control that defines the amount of compression to be applied. Points two to five below briefly discusses the workflow using Tab1 while point six elaborates on Tab2.
2. The user dynamically selects a reference audio file, i.e. a compressed audio file from which the DRC parameters are to be extracted. The reference file is dragged and dropped to the *list box* in the plugin and the path of the reference file is displayed in the interface of the plugin, as shown in Figure 1b.
3. The user then plays the track on which the the compression parameters are to be applied.
4. Once the *"Start learning"* button is pressed, a record functionality in the plugin begins to capture the live track playing in the digital audio workstation (DAW). The live audio stream is recorded for a duration of 2 seconds and written to a .WAV file at a predetermined location on the local machine. The path name of the reference audio file and recorded audio file is sent to a Python process on the local machine, where the raw audio waveform is fed into a pre-trained deep neural network model as input. The model outputs a set of parameters which are scaled and returned to the plugin, where they are used to configure the GUI of the plugin as shown in Figure 1c. At this stage, the entire process mentioned in these steps takes 4 to 5 seconds to complete on an average.
5. The user can now treat the estimated parameters as the seed parameters to work on. Instead of full automation, the user is allowed to tweak the parameters further.
6. The second tab is a single control interface as shown in Figure 1c and 1d, where steps 2-4 are repeated. Once the predicted and scaled parameters are received from the Python implementation of the DNN, the threshold is normalized and set on the compression knob of the GUI. The user can vary the threshold level using the knob, all other values remain unchanged given the model's prediction.

### 3.2 C++ to Python interfacing

Since the plugin was developed in C++ for real-time audio processing while the back-end model proposed in [7] was written in Python, we utilised Pybind[2] for interfacing. Pybind is a header only library that exposes C++ types in Python and vice versa.

Path names of the reference and recorded audio files selected in the DAW are sent from the plugin to the back-end Python process, where a short segment of both audio files are fed to a neural network model. Short characteristic excerpts from both audio files are selected because currently the deep neural network model accepts and processes a short (1s) excerpt for computational reasons. We can deal with longer term variation in the future using multiple segments for example.

<table>
<tr><td colspan="3">Input: Waveform (44100,1)</td></tr>
<tr><td rowspan="5">Front-end Network</td><td colspan="2">Conv1D: 3*1*64<br>Batch Normalisation</td></tr>
<tr><td>Conv1D: 3*1*64<br>Batch Normalisation<br>MaxPool1D: 3*1</td><td>*2</td></tr>
<tr><td>Conv1D: 3*1*128<br>Batch Normalisation<br>MaxPool1D: 3*1</td><td>*2</td></tr>
<tr><td>Conv1D: 3*1*256<br>Batch Normalisation<br>MaxPool1D: 3*1</td><td>*2</td></tr>
<tr><td colspan="2">Flatten, Dimension_expand</td></tr>
<tr><td rowspan="6">Back-end Network</td><td>Conv2D: 7*256*512<br>Batch Normalisation</td><td>L1;</td></tr>
<tr><td>Conv2D: 7*256*512<br>Batch Normalisation</td><td>L2;</td></tr>
<tr><td>Add (L1, L2)</td><td>L3;</td></tr>
<tr><td>Conv2D: 7*256*512<br>Batch Normalisation</td><td>L4;</td></tr>
<tr><td colspan="2">Add (L3, L4)</td></tr>
<tr><td colspan="2">Global Pooling;<br>Dense(feature embedding layer): 50<br>Dense: num_para</td></tr>
<tr><td colspan="3">Output: Parameters</td></tr>
</table>

Table 1: This is the architecture of the network for each branch of the siamese model proposed in [7]. It consists of 7 convolutional layers followed by batch normalisation and max pooling in the front end network, followed by two residual layers and one dense layer.

### 3.3 Model architecture and processing

The siamese model [30] is a neural network structure that contains two or more identical sub networks with

[2] https://github.com/pybind

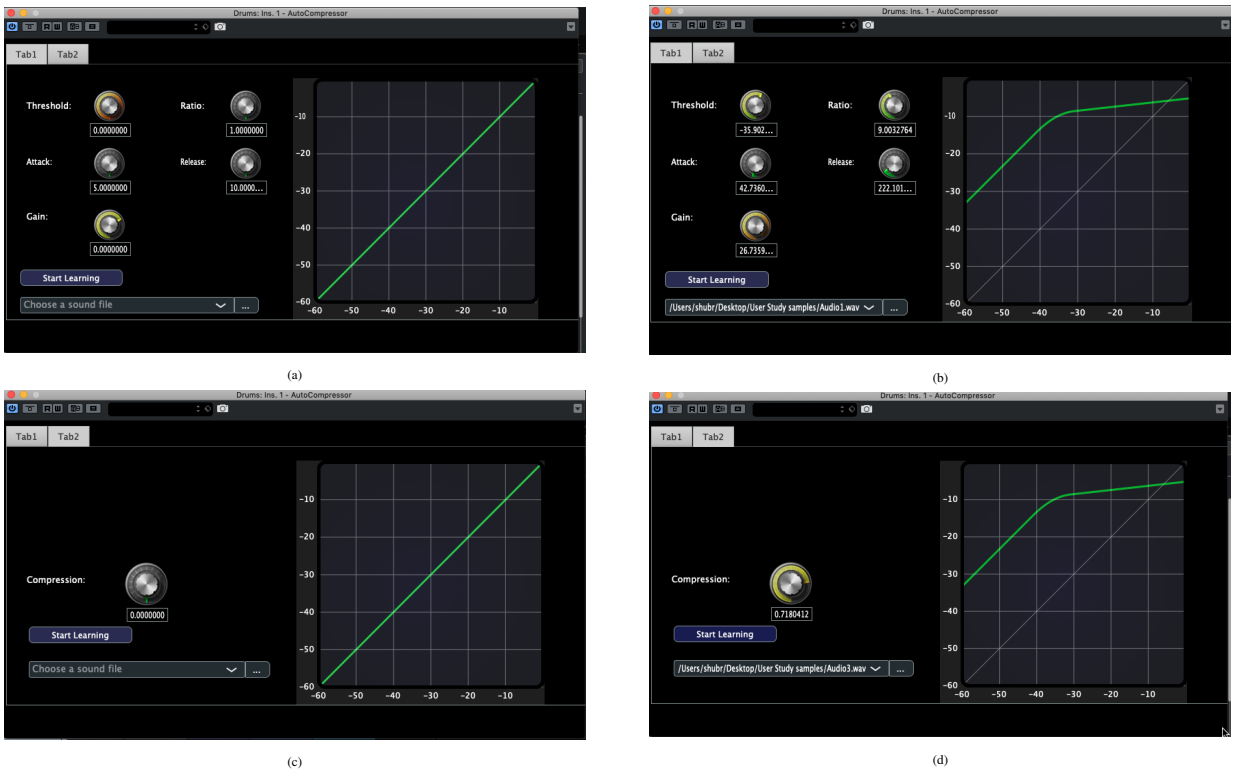


Figure 1: Screenshots of the automatic compressor plugin. 1(a) shows Tab1 of the plugin. 1(b) shows the display of the plugin after dragging or selecting the reference sound file. The model predicts the parameters after the *"Start Learning"* button is pressed. Predicted parameters are set on the GUI upon the completion of the process. Similar operation takes place for Tab2 in 1(c) and 1(d), where only a single control for *"compression amount"* is present.

shared weights. Siamese neural networks became increasingly common for learning a degree of similarity or learning to distinguish between related and unrelated items in computer vision and other domains. In our application, by designing an appropriate merge function, we can tune the model to pay attention to subtle changes in the input signals. One branch will "see" the compressed reference audio whereas the other would have the recorded track as input. For the purpose of the compressor plugin, we used the waveform input model proposed in [7]. As the name suggests, the model accepts raw time domain waveform as input instead of a more conventionally utilised time frequency representation.

The architecture of the network used in each branch of the model is summarised in Table 1. For better understanding, the model structure can be separated into a front-end and a back-end network. The front-end network consists of seven 1-D convolutional layers, batch normalisation and six layers of max pooling. This front-end is followed by two residual layers and one dense layer. The residual layers are utilised to help avoiding the "vanishing gradient" problem without introducing too many layers. Both branches output feature embeddings which are merged together using a subtract layer. The subtract layer is a custom Keras[3] layer which takes two tensors (input[0] & input[1]) of the same shape as input and returns a single tensor (input[1] - input [0]), also of the same shape. The subtraction layer is followed by a fully connected dense layer which outputs the predicted parameters. We use mean squared error as the loss function for training the network. Further discussion about the model and its training procedure can be found in [7] together with comparison to other network architectures and the assessment of hyper parameters of the model.

### 3.4 Makeup Gain calculation

The make up gain value is not currently predicted by the model and will be explored in future research. In the current implementation, gain is calculated by subtracting the EBU R128 [31] value of the reference audio from the recorded audio. This method is based on the ITU-R BS.1770 recommendation for broadcast loudness measurement.

## 4 Usability Study

In the following sections, we discuss the usability study conducted with the plugin to test the proposed DRC control mechanism in practice. Our primary focus is on evaluating the acceptance of the plugin in an operational environment with respect to different types of users. In our experiment, we deliberately avoid focusing on compression quality. The success of the proposed algorithm for parameter estimation and comparison between alternatives were discussed in previous work [7] while the method proposed for this study is outlined in Section 3.3.

### 4.1 Objectives

The primary objective of the study was to evaluate the likelihood of the proposed control mechanism for the DRC

[3] `https://keras.io/`

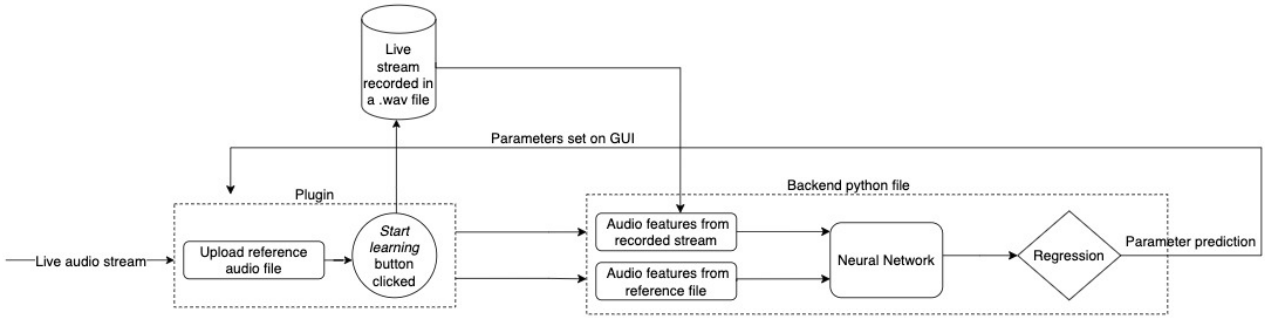


Figure 2: High level overview of the compressor's workflow. Live audio stream from the DAW is recorded and the path of both the recorded audio and reference audio is sent across to the back-end Python process. The binding between the C++ plugin and the Python implementation is done using Pybind. Once the path names are received, an excerpt is extracted from both audio files and fed into the network which outputs the predicted parameters. These parameters are sent back to the compressor plugin and the variables in the plugin code are set accordingly.

being used by different types of users in a music production workflow. The secondary objective was to informally assess the users' perception about performance, ease of use and the interface of the plugin.

### 4.2 Participants

Participants were recruited from Queen Mary University of London, The Academy of Contemporary Music in Guildford, and Steinberg Media Technologies GmbH, London. In total, 21 users took part in our study comprising of 19 male and 2 female participants. The test participants consisted of four self identified professional music producers, the rest were predominantly pro-amateur and amateur users of music technology [32]. Based on music production experience, we divided the users into two groups :

- Ten users with greater than 5 years experience in music production. This group was treated as the professional amateur (Pro-am) group for the quantitative evaluation. We used the term Pro-am because the group consists of full time professional music producers along with users who have a significant amount of experience but are not professional musicians.
- Eleven users with less than 5 years experience in music production. This group was treated as the amateur group for both quantitative and qualitative evaluation.

The aforementioned grouping structure was used while conducting quantitative evaluation, however for qualitative evaluation the Pro-am group was further split into Pro-am and professionals. The reason for this was that from the context of usability, preference metrics for professional users can vary significantly from amateur or Pro-am users. For example, the time saved due to automatic prediction of parameters may be a critical benefit for a professional user.

### 4.3 Audio Materials

For this study, five different sound samples created by the second author were used. They are Acoustic Bass; Acoustic Drums; two Electronic Drums and Synthesised Bass. All the samples were compressed using Cubase's built in compressor[4] Samples were recorded as WAV files and were both processed and presented to the participants at a sample rate of 44.1 kHz. These audio samples served as the corpus of reference audio files from which a user could select any file and upload it to the plugin for estimating the compression parameters. All files were loudness matched and normalised to -26 LUFS.

### 4.4 Procedure & Methodology

The study was conducted on the primary author's laptop. A high quality studio headphone by Beyerdynamic (DT-770 Pro) was used for all participants. We acknowledge that choosing headphones to conduct the test is a potential limitation. However, it was not practical to conduct the test under lab conditions, partly due to greater access to students in a classroom setting and professionals in their usual work environment. Generally, for an acceptance test in the workflow, we don't consider this a serious limitation. Other potential limitations are discussed in Section 6.

First, a briefing about the experiment was provided to each participant. This was followed by a demo of the plugin by the author where the participant could listen to any of the five reference audio recordings and select one of them for processing with the plugin. A synthetic drum loop was played using Cubase Pro 10 and the reference audio was selected in the plugin. Once the learning was complete and the parameters were set in the GUI of the plugin, the participant was asked to observe the perceptual change in the original track due to compression with the predicted parameters.

After the demo, users were asked to operate the plugin independently and explore at least 3 different reference audio examples and notice how close the synthetic drum loop comes to the selected reference audio.

### 4.5 Evaluation Methodology

A Participant Questionnaire was used to collect quantitative and qualitative data. For evaluating our primary objective, the following two questions were included in the questionnaire:

[4] https://new.steinberg.net/cubase/

1. Q1: Would you use such a plugin in your current workflow? (Options: Yes, No, Maybe)
2. Q2: Would you describe the operation of the plugin as intuitive? (Options: Yes or No)

These questions are designated Q1 & Q2 respectively for the sake of reference in the rest of this paper.

For evaluating our secondary objectives, rating and Likert-style questions as well as open-ended questions were used. The following questions were included in the questionnaire:

1. Q3: How easy / difficult was it to use the plugin? (Options: (1-7) Likert scale, ranging from extremely easy to extremely difficult)
2. Q4: How do you rate the automatic compression functionality of the plugin in terms of performance? (Options: (1-7) ordinal scale, ranging from very poor to excellent)
3. Q5: If your answer to Q1 is either No or Maybe, kindly provide further details.
4. Q6: What did you like about the plugin.

Q3 & Q4 were used for quantitative while Q5 & Q6 were used for qualitative analysis of the plugin. For Q5 & Q6, multi line text boxes were provided and the open ended responses from each user were collected for the qualitative analysis. Q5 was a mandatory question in case the answer to Q1 was No/Maybe. Statistical tests were selected according to the study design and distribution characteristics of the responses, while thematic analysis [33] was used for the qualitative aspects of the study.

## 5 Results

In this section, we briefly outline our findings from the quantitative and qualitative analyses. Critical assessment of these results are provided in Section 6.

### 5.1 Quantitative results

Responses to four questions (Q1-Q4) were analysed using statistical techniques to investigate the acceptance of the proposed workflow given the objectives detailed in Section 4.1, i.e., likelihood of using the proposed automation (primary), performance and ease of use (secondary).

1. Primary Objective Q1 - 45.45% (5 of 11 users) from the amateur group responded with *yes* to Q1 with 90% confidence interval ranging from 24.2% to 68.5%. In the Pro-am group, 70% users (7 of 10) responded with *yes* to Q1 with 90% confidence interval lying between 44.2% to 87.3%. Wilson score interval [34] was used to calculate the confidence interval.

2. Primary Objective Q2 - 100% (all 11 users) from the amateur group responded with *yes* to Q2 (rating the plugin intuitive) with 90% confidence interval ranging from 80.03% to 100%. For the pro-am group 100% users (10/10) said yes to Q2 with 90% confidence interval ranging from 78.7 to 100%.

3. In case of the secondary objective Q3 - A Mann Whitney test [35] indicated that the response from amateur group (mean = 1.72, mode = 2) is significantly different from the pro-am group (mean= 2.25, mode = 3). $U = 27$, $p = 0.018$ and $\alpha = 0.05$

4. Secondary objective Q4 - A Mann whitney test indicated that the response from amateur group (mean $= 5.09$, mode $= 4$) is significantly different from the pro-am group (mean $= 5.9$, mode $= 6$). $U = 31.5$ , $p = 0.047$ and $\alpha = 0.05$.

### 5.2 Qualitative results

A thematic analysis [33] was applied to the responses recorded through Q5 and Q6 of the questionnaire. Coding was performed by two authors and reviewed independently by another. Four main themes (T1-T4) emerged from the aforementioned analysis:

- T1: Intuitive and fast to use (+)
- T2: Closely matches the reference sound (+)
- T3: Impedes creative process (-)
- T4: Lack of information about how parameters are predicted (-)

The designation (+) indicates a positive theme, while (-) indicates criticism. For the thematic analysis, we rearranged the grouping used in quantitative analysis. Full time professional users were categorized in one group (professionals) and the rest were categorized in the amateur group. The total number of users in the professional group was 4, while the amateur group had 16 participants. As discussed earlier, the reason for this grouping can be justified by considering that self-identified professional who make a living from working with audio may have substantially different requirements and views on tools that affect their workflows, compared to Pro-am and amateur users who are more likely to engage in free exploration. Therefore we intend to analyse the self-identified professional group separately.

As shown in Figure 3, criticism of the plugin came from the amateur group, referring to (i) how the idea of parameter automation may impede the creative process (4/16 users), and (ii) indicating a lack of human understandable representation of the mechanism the model uses to predict the parameters (6/16 users). Some users considered this would make the interaction more engaging and useful for the them. These critical responses can be attributed to the fact that the amateur group included researchers with an interest in audio processing algorithms, and also to the larger number of participants in this group for the purposes of qualitative data analysis.

A positive theme emerging in the amateur group was related to our secondary objective, suggesting that the plugin was intuitive and the parameter prediction was fast (11/16 users). The second theme indicates that the plugin matches

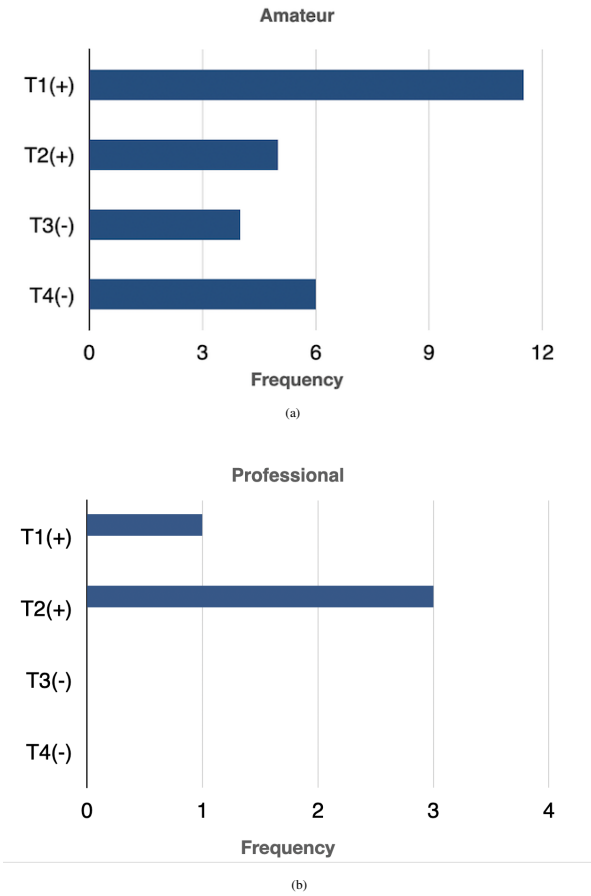


Figure 3: Thematic analysis for Amateur (3a) and Professional (3b) users. (+) refers to positive theme and (-) refers to negative theme.

the live track to the reference track to a good extent (5/16 users).

From the analysis of the professional group, it can be seen in Figure 3b that no criticism was received, while the two positive themes that emerged in the amateur group were consistent with the professional group as well.

## 6 Discussion

Based on the quantitative analysis, professional and pro-am users are more likely to use the type of automation introduced by our plugin in their daily workflows (Q1). Amateur users seem less certain about this choice however. Regarding the interaction with the plugin, professional, Pro-am and amateur users have predominantly found the process of selecting a reference audio and using automatically estimated parameters based on this intuitive. Regarding ease of use (Q3) opinions diverged between the user groups, with more experienced users finding it easier. This is possibly due to the fact that experienced users are more likely to have encountered plugins that include a "learning stage", such as a noise reduction plugin. There is divergence in the qualitative judgement of the plugin's performance, with experienced users rating it higher. This may be attributed to the fact that hearing differences in the dynamic range requires critical listening skills which develop with experience.

The qualitative analysis revealed that professional users, not surprisingly, are more demanding, with a smaller proportion commenting positively on the plugin's speed and intuitiveness. However, professional users generally considered the processed audio a good match to the reference in their feedback comments. Amateurs on the other hand appeared to be more curious about the internal working of the plugin and considered in higher proportion (4 of 16) that the automation may impede the creative process.

There are several aspects of the study that need to be taken into consideration before making claims about the usefulness of the compressor plugin. Firstly, the sample size in terms of participants is relatively small. The confidence interval of Q1 is quite wide for both groups indicating uncertainty in making a conclusive inference about the acceptability of the core concept of the plugin.

Another aspect to be taken into consideration is the potential biases introduced due to the design of the study. We acknowledge that the following points might have biased the participants:

- Since the capability of machine learning and artificial intelligence is discussed quite often on social media and news channels, the idea of a plugin with machine learning model might have biased some of the participants.
- The study was conducted in university campus and in the Steinberg office (for some professionals), which might not have been the preferred setting for some of the users to evaluate the plugin.
- Finally, some of the questions posed in the study might have been suggestive and could have biased the participants to provide a positive feedback about the plugin. For example, "Would you describe the operation of the plugin as intuitive?" Albeit, we consider the impact of this small and the questions sufficient for prototyping, especially on balance of conducting a more complex experiment that avoids self-reporting.

In the context of interpreting these results, it is important to mention here that the software is in prototype stage and the main purpose of the study was usability of the plugin and acceptance of the proposed type of automation in audio production workflows. Hence, qualitative evaluation had precedence over quantitative evaluation.

From the qualitative perspective, it was particularly encouraging to see a general consensus on the intuitiveness, ease of use and functionality of the plugin from both amateur and professional users. The two negative themes that emerged from thematic analysis also encourage us to devise new controls that allow the users to be more creative with the plugin. For example, a control to modify some parameters the neural network model so that new kinds of sounds can be produced could facilitate incidental exploration in a different way compared to manipulating the DRC parameters directly. Visualisation of what the model "pays attention to" when predicting the parameters would contribute to the explainability and transparency of the approach.

## 7 Conclusion and Future Work

This paper presented the development and user evaluation of an automatic compressor plugin utilising a control mechanism for the DRC using a reference audio signal. The plugin leverages a deep neural network model and a reference sound to infer the compression parameters. A user study was conducted with participants of varied music production background to evaluate the acceptability of the plugin in their work. Based on statistical analysis, professional users are somewhat more likely to incorporate the plugin in their music production workflow compared to amateur users. This can probably be attributed to their practice, such as the need for batch processing a large amount of content in certain situations and typical time constraints in production in professional settings. Based on the quantitative analysis of three questions as well as the thematic analysis, it can be inferred that users positively assessed the auto compression functionality and the operation of the plugin in general. It is interesting to note that while our motivation was partly to support amateur users, the study shows grater utility of the control mechanism using a sound example among professionals.

In a broader context of applications of the proposed approach to automation in general, and the plugin in particular, it will be valuable to investigate how our solution fits alongside semantic audio tools that use content analysis to facilitate workflow automation in audio and media production [36, 26, 27, 37, 38]. Further investigation into the context surrounding a workflow [2] and its relation to audio that is being processed would also be valuable. The relationships between context, audio features and audio effects may be represented using appropriate ontologies [39, 40, 41] and used to guide the parameter selection mechanism either through hyper-parameter optimisation or model selection.

From the perspective of future work, we intend to conduct the study with a larger sample population. We also intend to resolve the dependency on Python and make the implementation completely C++ based. This will enable deploying the plugin in more diverse work environments and across a larger user base for evaluation. The longer term road map includes applying the operation principle of the plugin in the context of different audio effects. We also consider improving the user experience given the qualitative feedback comments and address the needs of some users who seem to require more disclosure about how the processing parameters are estimated. This is likely to require innovation in HCI design as well as improving the interpretability of the neural network model.

## 8 Acknowledgements

The authors thank for the support and guidance from Jean-Baptiste Rolland and Yvan Grabit of Steinberg Media Technologies GmbH. A special thanks to Julien Tritsch (Steinberg Media Techonolgies GmbH, UK) for helping us recruit professional music producers for the user study and also for providing valuable ideas for the future work.

## 9 Bibliography

[1] D. Giannoulis, M. Massberg, J. Reiss, "Digital Dynamic Range Compressor Design—A Tutorial and Analysis," *Journal of The Audio Engineering Society* (2012).

[2] N. Lefford, G. Bromham, G. Fazekas, D. Moffat, "Context Aware Intelligent Mixing Systems," *Journal of the Audio Engineering Society* (2020), doi:https://doi.org/10.17743/jaes.2020.0043.

[3] Z. Ma, B. De Man, P. Pestana, D. Black, J. Reiss, "Intelligent Multitrack Dynamic Range Compression," *Journal of the Audio Engineering Society. Audio Engineering Society* (2015), doi:https://doi.org/10.17743/jaes.2015.0053.

[4] S. Gorlow, J. D. Reiss, "Model-Based Inversion of Dynamic Range Compression," *Trans. Audio, Speech and Lang. Proc.* (2013), doi:10.1109/TASL.2013.2253099.

[5] R. Stables, S. Enderby, B. De Man, G. Fazekas, J. Reiss, "SAFE: A system for the extraction and retrieval of semantic audio descriptors," presented at the *In 15th International Society for Music Information Retrieval Conference (ISMIR* (2014).

[6] D. Sheng, G. Fazekas, "Automatic control of the dynamic range compressor using a regression model and a reference sound," presented at the *Proceedings of the 20th International Conference on Digital Audio Effects (DAFx-17)*.

[7] D. Sheng, G. Fazekas, "A Feature Learning Siamese Model for Intelligent Control of the Dynamic Range Compressor," presented at the *Proc. of the International Joint Conf. on Neural Networks (IJCNN)* (2019), doi:10.1109/IJCNN.2019.8851950.

[8] W. Lee, S. Wang, F. Chen, X. Lu, S. Chien, Y. Tsao, "Speech Dereverberation Based on Integrated Deep and Ensemble Learning Algorithm," presented at the *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018), doi:10.1109/ICASSP.2018.8462662.

[9] D. Arifianto, M. Farid, "Dereverberation binaural source separation using deep learning," *The Journal of the Acoustical Society of America* (2018), doi:10.1121/1.5067488.

[10] R. Izhaki, *Mixing Audio: Concepts, Practices and Tools* (Focal Press) (2008).

[11] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, K. Simonyan, "Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders," presented at the *Proceedings of the 34th International Conference on Machine Learning* (2017).

[12] S. H. Hawley, B. Colburn, S. I. Mimilakis, "Profiling Audio Compressors with Deep Neural Networks," presented at the *Audio Engineering Society Convention 147* (2019).

[13] G. R. Koch, "Siamese Neural Networks for One-Shot Image Recognition," presented at the *ICML* (2015).

[14] P. Nick, "The rise of the new amateurs: Popular music, digital technology, and the fate of cultural production/Nick Prior," in *Handbook of Cultural Sociology* (2010).

[15] B. De Man, J. Reiss, "A Semantic Approach To Autonomous Mixing," *Journal of the Art of Record Production* (2013).

[16] E. Chourdakis, J. Reiss, "A Machine-Learning Approach to Application of Intelligent Artificial Reverberation," *Journal of the Audio Engineering Society* (2017), doi:10.17743/jaes.2016.0069.

[17] M. Martinez Ramirez, E. Benetos, J. Reiss, "Deep Learning for Black-Box Modeling of Audio Effects," *Applied Sciences* (2020), doi:10.3390/app10020638.

[18] E.-P. Damskägg, L. Juvela, V. Välimäki, "Real-Time Modeling of Audio Distortion Circuits with Deep Learning," (2019), doi:10.3390/app10030766.

[19] J. Covert, D. Livingston, "A vacuum-tube guitar amplifier model using a recurrent neural network," (2013), doi:10.1109/SECON.2013.6567472.

[20] T. Schmitz, J. J. Embrechts, "Nonlinear Real-Time Emulation of a Tube Amplifier with a Long Short Term Memory Neural-Network," (2018), doi:10.13140/RG.2.2.31008.28167.

[21] C. Steinmetz, J. Pons, S. Pascual, J. Serrà, "Automatic multitrack mixing with a differentiable mixing console of neural audio effects," (2020).

[22] C. J. Steinmetz, J. D. Reiss, "Efficient Neural Networks for Real-time Analog Audio Effect Modeling," (2021).

[23] A. J. R. Simpson, G. Roma, M. D. Plumbley, "Deep Karaoke: Extracting Vocals from Musical Mixtures Using a Convolutional Deep Neural Network," *CoRR* (2015), doi:10.1007/978-3-319-22482-4_50.

[24] S. Mimilakis, K. Drossos, T. Virtanen, G. Schuller, "Deep Neural Networks for Dynamic Range Compression in Mastering Applications," presented at the *Audio Engineering Society Convention 140* (2016).

[25] E. Vickers, "Automatic Long-term Loudness and Dynamics Matching," (2001).

[26] T. Wilmering, G. Fazekas, M. Sandler, "High level semantic metadata for the control of multitrack adaptive audio effects," presented at the *133rd Convention of the Audio Engineering Society, San Francisco, CA, USA* (2012).

[27] R. Stables, B. De Man, S. Enderby, J. Reiss, G. Fazekas, T. Wilmering, "Semantic description of timbral transformations in music production," presented at the *ACM Multimedia, Oct. Amsterdam, Netherlands* (2016), doi:10.1145/2964284.2967238.

[28] D. Giannoulis, M. Massberg, J. D. Reiss, "Parameter automation in a dynamic range compressor," *Journal of the Audio Engineering Society* (2013).

[29] B. De Man, J. D. Reiss, R. Stables, "Ten Years of Automatic Mixing," presented at the *Proceedings of the 3rd Workshop on Intelligent Music Production* (2017).

[30] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, R. Shah, "Signature Verification Using A "Siamese" Time Delay Neural Network." *IJPRAI* (1993), doi:10.1142/S0218001493000339.

[31] M. K. Højlund, M. S. Riis, D. Rothmann, J. R. Kirkegaard, "Applying the ebu r128 loudness standard in live-streaming sound sculptures," presented at the *NIME* (2017).

[32] M. Sandler, D. De Roure, S. Benford, K. Page, "Semantic web technology for new experiences throughout the music production-consumption chain," presented at the *2019 International Workshop on Multilayer Music Representation and Processing (MMRP)*, pp. 49–55 (2019), doi:10.1109/MMRP.2019.00017.

[33] V. Braun, V. Clarke, N. Hayfield, G. Terry, *Thematic Analysis* (Springer Singapore) (2019).

[34] E. B. Wilson, "Probable Inference, the Law of Succession, and Statistical Inference," *Journal of the American Statistical Association* (1927), doi:https://doi.org/10.2307/2276774.

[35] H. B. Mann, D. R. Whitney, "On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other," *The Annals of Mathematical Statistics* (1947), doi:http://dx.doi.org/10.1214/aoms/1177730491.

[36] G. Fazekas, M. Sandler, "Intelligent editing of studio recordings with the help of automatic music structure extraction," presented at the *122nd Convention of the Audio Engineering Society, Vienna, Austria* (2007).

[37] F. Font, T. Brookes, G. Fazekas, M. Guerber, A. La Burthe, D. Plans, M. Plumbley, W. Wang, X. Serra, "Audio Commons: Bringing Creative Commons Audio Content to the Creative Industries," presented at the *61st AES International Conference on Audio for Games, Feb 10–12, London, UK* (2016), doi:http://www.aes.org/e-lib/browse.cfm?elib=18093.

[38] C. Baume, *Semantic Audio Tools for Radio Production, Ph.D. thesis,* (University of Surrey) (2018).

[39] A. Allik, G. Fazekas, M. Sandler, "An Ontology for Audio Features," presented at the *17th International Society for Music Information Retrieval (ISMIR-16) conference, August 7-11., New York, USA*, pp. 73–79 (2016).

[40] T. Wilmering, G. Fazekas, M. Sandler, "Semantic Metadata for Music Production Projects," presented at the *Proc. of the 12th International Semantic Web Conference (ISWC), first International Workshop on Semantic Music and Media (SMAM2013)*, pp. 21–25 (2013).

[41] T. Wilmering, G. Fazekas, M. Sandler, "The Audio Effects Ontology," presented at the *Proc. of the 14th International Society for Music Information Retrieval Conference, ISMIR'13, November 4-8, Curitiba, Brazil* (2013).