Master's thesis

2022

Bjørn Are Therkelsen

NTNU
Norwegian University of
Science and Technology
Faculty of Information Technology and Electrical
Engineering
Department of Computer Science

Bjørn Are Therkelsen

Automatically Quantifying the Expressive Style of Piano Performances

May 2022

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Automatically Quantifying the Expressive Style of Piano Performances

Bjørn Are Therkelsen

Computer Science
Submission date: May 2022
Supervisor: Rune Sætre

Norwegian University of Science and Technology
Department of Computer Science

Abstract

This master's thesis explores how to distinguish pianists based on their expressive style. The styles are quantified based on extracted Mid-Level Perceptual Features (MLPFs) from piano performances. A new dataset of MLPFs was collected using musical domain experts, and ground truths were calculated based on the distributions of the experts' labels. Recurrent Neural Networks (RNNs) were trained to predict MLPFs and recognize individual pianists. The RNNs use sequences of played notes, automatically aligned to the original scores, as input.

The performance of the RNNs on the MLPF prediction task was evaluated using R^2 (coefficient of determination) scores. Even though the ground truths were based on the labels of domain experts, RNN models got closer to the ground truths than the experts. However, both the models and the experts had negative R^2 scores. RNN-based models were able to recognize individual pianists with an accuracy of 82.8%. The recognition accuracy was reduced by at least 7.5% when the MLPFs were used as an intermediate step. Calculating the interclass correlations of the dataset shows that only a few of the MLPFs for piano performance styles are considered reliable. This indicates that the new dataset of MLPFs is too small and subjective to train models to predict MLPFs. Extra collection of data and an MLPF prediction competition will try to solve these problems.

Keywords: Recurrent neural network, explainable artificial intelligence, expressive piano style

Sammendrag

Denne masteroppgaven utforsker hvordan man kan skille forskjellige pianister basert på deres stiler. Stilene kvantifiseres ved hjelp av sansbare egenskaper (såkalte Mid-Level Perceptual Features, MLPF) i opptak fra enkeltsanger. Et nytt datasett med MLPF-er ble samlet inn ved hjelp av musikalske domeneeksperter, og gullstandarder ble beregnet basert på gjennomsnittlige fordelinger av ekspertenes svar. Rekurrente nevrale nettverk (RNN-er) ble opplært til å predikere MLPF-ene og til å gjenkjenne individuelle pianister. RNN-ene bruker sekvenser av preprosesserte noter som er blitt automatisk justert mot de opprinnelige nedskrevne notene.

Ytelsen til RNN-baserte modeller på MLPF-prediksjonsoppgaven ble evaluert ved å bruke R^2 score. Selv om gullstandarden var basert på svarene til domeneeksperter, kom RNN-modellene nærmere gullstandarden enn ekspertene selv. Imidlertid fikk både modellene og ekspertene negative R^2 verdier. De RNN-baserte modellene var i stand til å gjenkjenne individuelle pianister med en nøyaktighet på 82,8%. Gjenkjenningsnøyaktigheten ble redusert med minst 7,5% når MLPF-ene ble introdusert som et mellomtrinn i beregningsprosessen. Beregning av interklassekorrelasjonene i MLPF-datasettet viser at bare noen få av MLPF-ene anses som pålitelige. Dette antyder at dette nye datasettet med MLPF-er er for lite eller at svarene er for subjektive til å trene modeller til å forutsi MLPF-er. Ytterligere datainnhenting og en kommende prediksjonskonkurranse vil prøve å løse disse utfordringene.

Nøkkelord: Rekurrente nevrale nettverk, forklarende kunstig intelligens, pianostil

Preface

Due to Covid-19, I could not go on exchange before the final year of my master's degree. Thus, I went to Seoul National University (SNU) in Seoul, Republic of Korea, while my supervisor was at the Norwegian University of Science and Technology (NTNU) in Trondheim, Norway. Without the help of my supervisors at SNU and NTNU, the writing of this thesis would not have been possible.

My background is more related to the computer science part of the project than the music part. I play the guitar and the ukulele, but I have never been a proficient pianist. My understanding of notes and audio formats has been developed through school and various side projects. However, my last five years at NTNU have been dedicated to understanding computer data and its application. Primarily the domain of artificial intelligence (AI) has been interesting. The methods of AI can automate human tasks and aid human decision processes. Yet, the study of artificial intelligence has highlighted many weaknesses of such techniques. For example, a machine cannot learn patterns if the data is not presented well enough or if the wrong method is used.

The topic of this thesis is interesting because people can relate to musical style without being able to describe it. Many expressions used to explain music are subjective and impossible to measure objectively. The domain of piano provides an excellent basis for researching expressive style. Pianists can be recorded using electronic pianos, and musical data can be extracted. Notes in MIDI format are clearly defined by onset and offset times. Additionally, the notes include information about loudness and pedaling can be measured. Connecting such low-level data to terms that are relatable to humans is complex. However, that only encouraged me to try to solve it.

I wrote the thesis mainly to extend the knowledge of objective measurements for expressive style. Therefore, the research is primarily targetting researchers interested in obtaining knowledge in that specific subdomain. The methods presented in the thesis are particular to piano music, but the ideas are transferable to other domains. The use of deviations from the average performer can, for instance, be relevant for different branches of art studies. Also, the research can be interesting for people who only understand either the musical or the machine learning domains. This thesis could broaden the perspective of their fields.

I want to thank my supervisor at NTNU, Asc. Prof. Rune Sætre. Moving to Korea would not have been possible without his flexibility. Attending SNU has broadened my perspective in terms of research and artificial intelligence. Additionally, living in Seoul has taught me about cultural differences. Asc. Prof. Sætre's input has improved my scientific writing and thinking. I also want to convey my appreciation to my two supervisors at SNU: Prof. Wen-Syan Li and Prof. Jong Hwa Park. The research project would not have been successful without Prof. Li's determination and proactive actions. He helped me understand the methods of explainable AI and the machine learning process. Prof. Park provided the musical input that the research project depended on. His willingness to learn about computer science was inspiring. We had many engaging discussions to optimize the approaches for both the musical and computer

science domains. Finally, I want to thank the group members of the Music XAI team. The graduate students Jisoo Park and Angela Weihan were crucial to the results of our research.

Bjørn Are Therkelsen, May 27th 2022, Seoul, Republic of Korea

Contents

List of Figures

List of Tables

Acronyms

ASHA Asynchronous Successive Halving. 10

CA Cortial Algorithm. 28, 32

CENS Chroma Energy Normalized Statistics. 27

CMA-ES Covariance Matrix Adaption Evolution Strategy. 10

CNN Convolutional Neural Network. 25–28, 31

CP Compound Word Representation. 26, 28, 32

CRP Chroma Discrete Cosine Transform-Reduced Log Pitch. 24

CVAE Conditional Variational AutoEncoder. 23, 29

CVRNN Conditional Variational Recurrent Neural Network. 25

DT Decision Tree. 23

DTW Dynamic Time Warping. 24

FFNN Feed Forward Neural Network. 24

GMM Gaussian Mixture Model. 25, 28, 30

GNB Gaussian Naive Bayes. 23

GRU Gated Recurrent Unit. V, VIII, 6, 9, 27, 40

HAN Hierarchical Attention Network. 23

HMM Hidden Markov Model. 23

ICC Intraclass Correlation. 35, 71

IOI Inter Onset Interval. 37

k-NN k-Nearest Neighbors. 23

KDE Kernel Density Estimation. 25, 30, 37, 83

LDA Linear Discriminant Analysis. 28

LSTM Long Short Term Memory. V, VIII, 6–9, 23–28, 40, 83

MER Music Emotion Recognition. 15, 33

MFCC Mel Frequency Cepstral Coefficient. 26, 27

MIR Music Information Retrieval. 14, 32

MLPF Mid-Level Perceptual Feature. I, VI, VIII, IX, XII, 1, 2, 35, 39–42, 45–49, 51, 57, 59–61, 71, 72, 75–81, 83, 84

N2M Notes-to-MLPFs. 2, 83

N2M2P Notes-to-MLPFs-to-Pianists. 2, 83

N2MP Notes-to-MLPFs-and-Pianists. 2, 83

N2P Notes-to-Pianists. 2, 83

NLP Natural Language Processing. 26

OTD Offset Time Duration. 37

REMI Revamped MIDI Derived Events. 26–28, 32

RF Random Forest. 23

RNN Recurrent Neural Network. 24, 39, 40, 43

SLR Systematic Literature Review. V, IX, 11, 15, 16, 23, 29

SNU Seoul National University. 35, 84

SQ Study found by Query search. 13, 17–21, 23, 26–32

SR Study found by Recommendation. 13, 18–21, 23, 26, 29, 31

SS Study found by the Snowballing method. 13, 19–22, 24, 25, 28–30, 32, 33

STFT Short-Time Fourier Transform. 27

SVM Support Vector Machine. 23, 27, 28

TPE Tree Parzen Estimator. 10

VAE Variational AutoEncoder. 27

ZCR Zero Crossing Rate. 27

1 Introduction

This chapter will introduce the domain and the research questions of the thesis. The scope of the musical domain will be reduced to the performance style of the piano. The research questions are related to a set of mid-level perceptual features, which are used to quantify the style of pianists.

1.1 Background and Motivation

Humans partake in activities that do not relate to the survival of the species, like arts and sports. In contrast to sports, art is not based upon explicit competition with objective measurements. Art is all about human expression and giving meaning to life. Artists or pieces of art are however constantly being compared to each other. Such comparisons can for instance be in the form of price tags, popularity among fans, or respect from other artists. In the domain of piano, a major part of the attention is paid to the interpretation of previously composed music. This attention can be linked to the domain's long history and great respect for the classical. The interpretation of piano music is an implicit competition, where the styles of well-known pianists are analyzed and compared. Even though the comparisons are subjective in nature, there are commonly accepted ideas of what is considered a good performance. Using machines to evaluate performances could make the evaluation more objective. This, however, requires machines to be able to capture the individuality of each pianist and convey the information to humans.

Aljanaki and Soleymani described the three levels of complexity regarding music [1]. The lowest level consists of notes, chords and other basic building blocks. Mid-level features, like tonal and rhythmic stability, are a set of concepts based on those lower-level blocks. According to Aljanaki and Soleymani, the mid-level features are subjective and can be difficult to define clearly [1]. An MLPF is a term commonly used to describe a mid-level feature that is perceptible to humans. The highest level of complexity is based upon the lower levels. Some high-level features are mood, genre and style.

Chowdhury et al. introduced an approach for explainable emotion recognition [2]. They used MLPFs as an intermediate step to explain the classifications. Three architectures were compared by their performance on the prediction task of MLPFs and the classification of emotions: Audio-to-Emotion (A2E) classified emotion directly from audio spectrograms. A2Mid2E used audio spectrograms to predict MLPFs, and MLPFs to predict emotions. The models for each of these steps were trained independently. A2Mid2E-Joint had the same architecture as the A2mid2E, but the models were trained jointly.

1.2 Goals and Research Questions

This Master's Thesis goal was to explore how to extract MLPFs from piano performances in order to quantify the expressive style of the pianists. To achieve this goal, methods from the domain of machine learning was used.

The architectures of Chowdhury et al. [2] was be modified to fit the challenge of pianist style. Notes-to-Pianists (N2P) and Notes-to-MLPFs (N2M) recognized pianists and predicted MLPFs, respectively. Notes-to-MLPFs-to-Pianist (N2M2P) and Notes-to-MLPFs-and-Pianist (N2MP) recognized pianists and predicted MLPFs, by training separately (N2M2P) or jointly (N2MP). The architectures were tested by a prediction task of MLPF data and a recognition task of pianists. MLPFs was hypothesized to provide a good representation of the style of pianists if a model was able to perform well on both tasks.

RQ1: *How can mid-level perceptual features be automatically extracted from piano performances?*

Because of the difficulty of clearly defining MLPFs, algorithms for automatic extraction are difficult to develop. The fact that humans can observe MLPFs, makes the data-driven approach a possible option. However, the subjective nature of MLPFs introduces the problem of inconsistent labeling of data. The MLPFs build upon lower-level features that must be extracted first. According to Stamatatos, extracting the deviations from the norm performance provides a better representation of a pianist's individuality than the pianist's deviation from the score [3]. Even though the method of using deviations from the average performance might produce better results, it requires additional retrieval of data in order to compute the average performance.

RQ2: *What is the impact of introducing mid-level perceptual features as an intermediate step, when recognizing individual pianists?*

Aljanaki and Soleymani stated that the MLPFs are important when attributing high-level features [1]. A high-level feature such as performance style should at least be explained partially by the MLPFs. Chowdhury et al. computed the cost of using MLPFs as an intermediary step for recognition [2]. The cost was called the cost of explainability because the intermediary step of MLPFs provideed explainable features, but reduced the accuracy of the recognition task. Explainable features can make humans trust machine learning models since the predictions can be accompanied by reasons for the predictions. Better explainability will therefore have a positive impact on the overall task. On the other hand, a decrease in accuracy entails a negative impact.

2 Background Theory

This chapter provides a basic framework of musical and machine learning fundamentals to discuss this thesis's research questions. The musical part presents concepts linked to the domain of piano. The section regarding machine learning will introduce the concept of recurrent neural networks. Models based on this architecture are naturally suited for sequential data and will be used to extract the style from performances.

2.1 Music

The musical fundamentals include explaining a few key concepts: Score, timing and pitch are central to the understanding of piano music. Different aspects of musical style are also presented. The scope of this thesis is only set to one of the mentioned aspects of musical style: performance style.

2.1.1 Score

Music is typically defined in a score, using symbols to specify pitch and timing [4]. A musician can read these symbols and recreate the music of the original creator, called the composer. The score can be represented physically or digitally and in formats that are easy to understand for humans and machines. People typically prefer using staff with notes, as we can see in [4]. Standard MIDI File is often used for devices. This file type has a hierarchical XML structure, making it possible for people to understand its content. Users can easily find supported packages for various programming languages and publicly available musical data sets in this format.

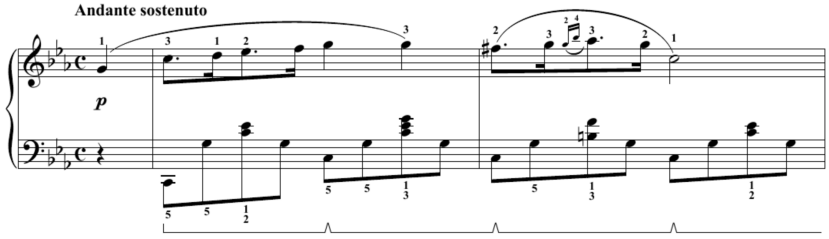


Figure 1: Score Excerpt from Nocturne in C Minor by Frédéric Chopin

2.1.2 Timing

Rhythm refers to the organization of musical elements in time [5]. The time is divided into beats and the tempo defines how many beats a specific time interval should include. Beats, therefore, define relative relationships in time, where the tempo can be used to adjust the distances when playing. The fundamental building block of musical structures is a note, representing a sound played for a given duration. These are often combined in a hierarchical manner, where

the lowest level is called a measure (or a bar). The time signature is the score component that relates the beats and the notes to each other. The top number represents the number of beats per measure, and the lower number represents the beat value attributed to each note (see Figure 2). The onset- and offset values describe a note's start- and end times in a MIDI file. These values are measured seconds after the beginning of the performance.

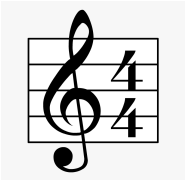


Figure 2: Time Signature

2.1.3 Pitch

The pitch of a note is how low or high it sounds. The pitch is related to the physical wavelength of a given sound, where a shorter wavelength (higher frequency) corresponds to a higher sound. In *Western Musical Notation*, the pitches are designated by alphabet letters: A, B, C, D, E, F, G (or sometimes solfège syllables: Do Re Mi Fa Sol La Ti) [4]. In the score, the pitches are represented by the vertical placement of a note on the staff. Figure 3 shows the commonly used staff, with five lines and four spaces. On the left side of the staff we can see the treble clef. This symbol defines the relationship between the lines of the staff and the scale of pitches, designated by the alphabet letters.

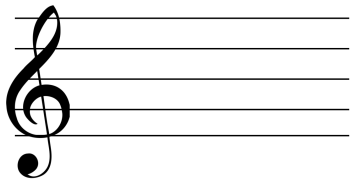


Figure 3: Staff with a Treble Clef

In the case of a treble clef, it places the alphabetic scale on the staff such that the second lowest line corresponds to a *G*. Figure 4 shows the alphabetic scale of pitches on top of the staff, given a treble clef.

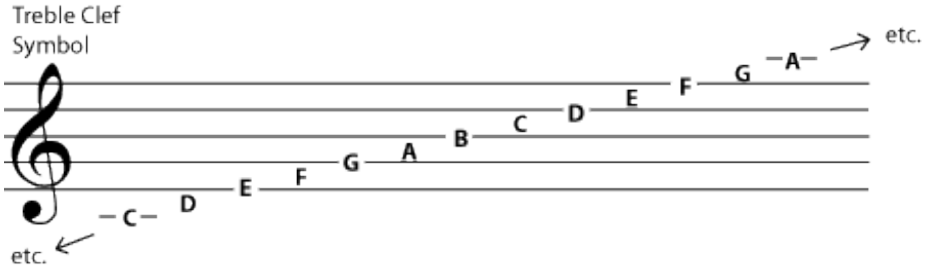


Figure 4: Musical Scale on the Staff

2.1.4 Style

Dannenberg discussed how the term style is commonly used in the music domain [6]. First, style can be associated with historical periods, such as the Baroque or the Classical period. Dannenberg lists a few characteristics connected to each of these periods: Contrapuntal and ornamented is typical for Baroque, and homophonic and internal structure is often associated with the Classical. The difference between contrapuntal and homophonic concerns the combination of multiple voices. Ornamentation is about inserting short, fast notes above or below a note of the melody. Ornamentation can be understood as the decoration of the melody. The classical style is plainer and focuses on developing formal structure. Both of these dimensions are linked to the composition of the pieces. A style attributed to a historical period explains the commonalities of the composers of that period. In this case, the style concerns a piece's structure, not its interpretation.

Next, style can also be associated with improvising performers. Dannenberg stated that especially improvising performers have their own styles, like the ballad style of Miles Davis [6]. In the case of improvisation, performers are composing and executing simultaneously. That entails that the style concerns a combination of the two.

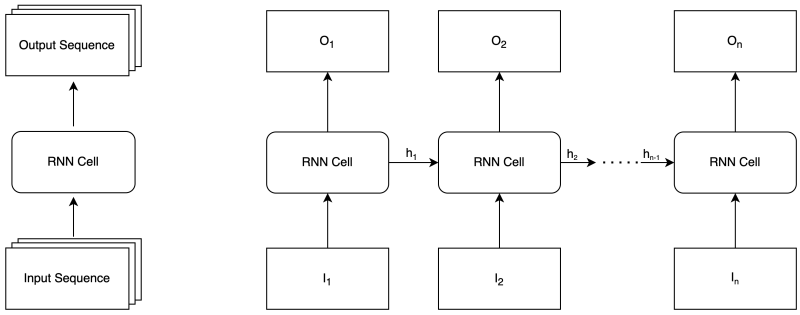
Finally, Dannenberg mentioned that expressive style could be associated with performers [6]. Expressive style is linked to situations where the style involves only the interpretation of a piece. The piece is already composed, so the performer has a limited number of ways to impact how the music will sound. For example, in the piano case, the performer can vary aspects such as the start time, the duration and the loudness of notes. Even though these aspects are defined in the score, the performer still has much room for interpretation. In addition, some instruments have style aspects that are not defined in the score. String instruments can, for instance, add vibrato to change the style of the performance.

2.2 Machine Learning

This section introduces the machine learning methods used in this thesis. Extracting mid-level features from note information is based on recurrent neural networks (RNNs). The recurrent neural networks use cells to store states and compute outputs for sequential data iteratively. Long ShortTerm Memory (LSTM) and Gated Recurrent Unit (GRU) are two examples of such RNNs, which will be used in this thesis. Optuna is a tool for optimizing hyperparameters. This tool is used when training the RNN models to produce optimal results on the machine learning tasks.

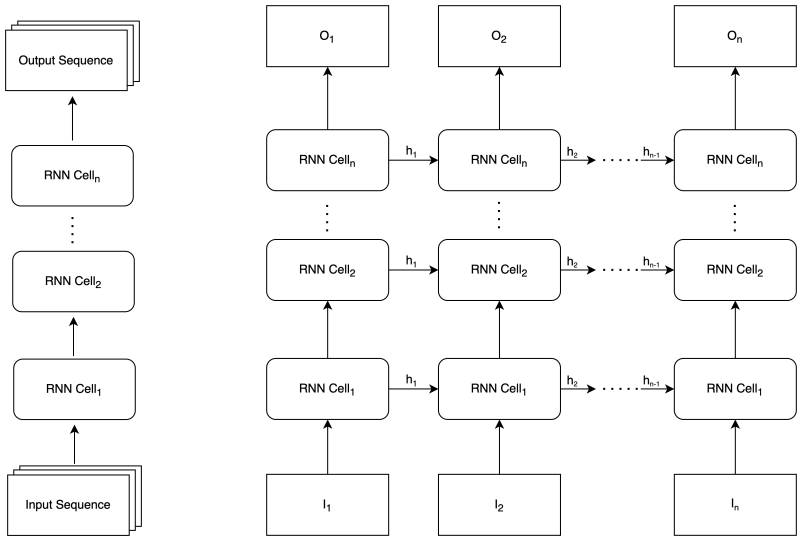
2.2.1 Recurrent Neural Network (RNN)

Recurrent neural network denotes a class of machine learning models that maps sequences of input data to sequences of output data (see Figure 5a). An RNN comprises cells that compute the output based on the input and a hidden internal state. These cells are used sequentially, where the output (O_t) is calculated and the hidden state (h_t) is updated for each individual input (I_t) in the input sequence (Figure 5b). The basic version of an RNN uses only one cell to compute the output sequence. It is also possible to stack RNN cells such that the output of one cell is used as input for the next cell (Figure 6). The input sequences can be variable in length since the output sequence's length will match the input sequence's length. DiPietro and Hager state that recurrent neural networks are naturally suited to process time-series and other sequential data [7].



(a) Sequence (b) Individual

Figure 5: RNN Architecture with a Single Cell



(a) Sequence (b) Individual

Figure 6: RNN Architecture with Stacked Cells

2.2.2 Long Short-Term Memory (LSTM)

LSTM is an RNN cell developed to solve the vanishing gradient problem. This problem will be briefly summarized based on the explanations of DiPietro and Hager [7]. When computing the gradients of an RNN cell, the change in output value concerning the change in input value ($\frac{\delta^+ h_T}{\delta \theta}$) is based on the sum of all sequential steps (Equation 1).

$$\frac{\delta^+ h_T}{\delta \theta} = \sum_{t=0}^{T-1} \frac{\delta^+ h_T}{\delta^+ h_{T-t}} \frac{\delta^+ h_{T-t}}{\delta \theta} \tag{1}$$

The change in output for time step T with respect to the change in output for time step t ($\frac{\delta^+ h_T}{\delta^+ h_{T-t}}$) is the product of a sequence of matrices (Equation 2).

$$\frac{\delta^+ h_T}{\delta^+ h_{T-t}} = \frac{\delta^+ h_T}{\delta^+ h_{T-1}} \frac{\delta^+ h_{T-1}}{\delta^+ h_{T-2}} ... \frac{\delta^+ h_{T-t+1}}{\delta^+ h_{T-t}} \tag{2}$$

If the spectral norm of the matrices is less than 1, the contributions to the gradients from events before the last time step will fall off exponentially. Therefore, the sequential aspect of the RNN is diminished since the gradients are computed based on only the last few time steps. The goal of the LSTM cell is to provide stable spectral norms so that the gradients are based on the all-time steps of the input sequence. Stable spectral norms are achieved by adding additional paths between the hidden states. The inner computations of an LSTM cell from the Pytorch framework [8] is expressed in formulas in Equation 3 and visualized in

Figure 7. The f_t, i_t, g_t and o_t denote the forget-, input-, cell- and output gates at time step t respectivelly. x_t, h_t and c_t correspond to the input- hidden- and cell states. The output for a cell is the hidden state, h_t. The Hadamard product corresponds to $\odot$ and σ is the sigmoid function. The weights, W_{xy}, refers to the trainable parameters for the xth state (i for input state, or h for hidden state) and the yth gate (f, i, g, o). The biases, b_{xy}, are denoted in the same way.

$$
\begin{aligned}
f_t &= \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \\
i_t &= \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \\
g_t &= tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \\
o_t &= \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \qquad (3)
\end{aligned}
$$

$$
\begin{aligned}
c_t &= f_t \odot c_{t-1} \odot g_t \\
h_t &= o_t \odot tanh(c_t)
\end{aligned}
$$

The visualization in Figure 7 uses the same annotations as in Equation 3. WSB_x is added as an abbreviation for weigthed sum and add biases for gate x.

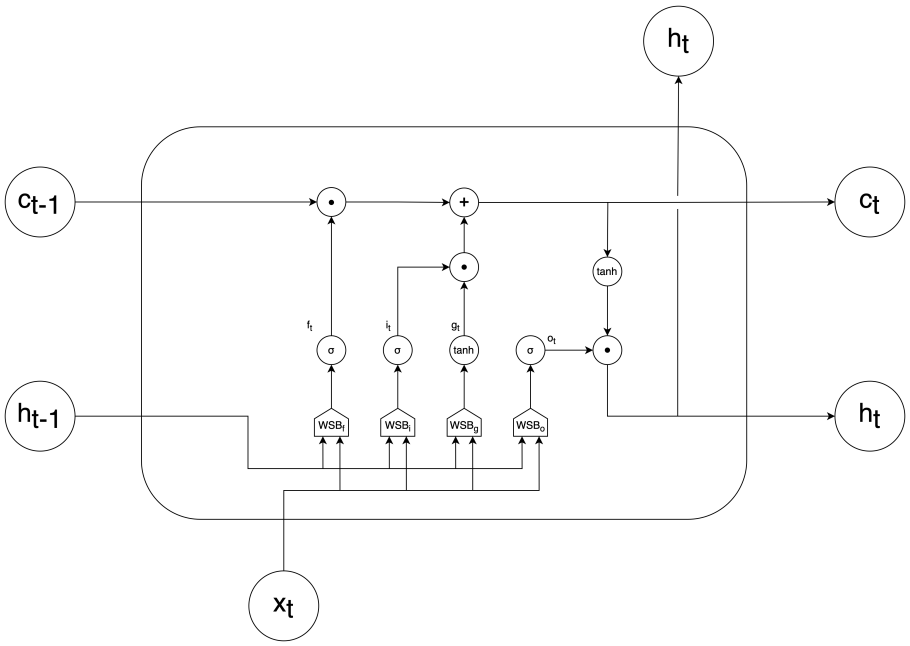


Figure 7: LSTM Cell

2.2.3 Gated Recurrent Unit (GRU)

GRU was created to simplify the LSTM cell. The GRU architecture provides stable spectral norms, while not using a separate cell state. The inner computations of the GRU cell from the Pytorch framework [8] are expressed in formulas in Equation 4 and visualized in Figure 8. As in the case of the LSTM, x_t and h_t correspond to the input- and hidden states at time step t. The r_t, z_t and n_t denote the reset-, update- and new gates at time step t. Weights, biases, operators and functions are represented in the same way as for the LSTM cell.

$$
\begin{aligned}
r_t &= \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{t-1} + b_{hr}) \\
z_t &= \sigma(W_{iz}x_t + b_{iz} + W_{hz}h_{t-1} + b_{hz}) \\
n_t &= \tanh(W_{in}x_t + b_{in} + r_t \odot (W_{hn}h_{t-1} + b_{hn})) \qquad (4) \\
\\
h_t &= n_t \odot (1 - z) + z_t \odot h_{t-1}
\end{aligned}
$$

In addition to the annotations used in the visualization of the LSTM cell, WB_x is added to Figure 8 to denote the weighting of the xth state and adding biases for the new gate (n_t).

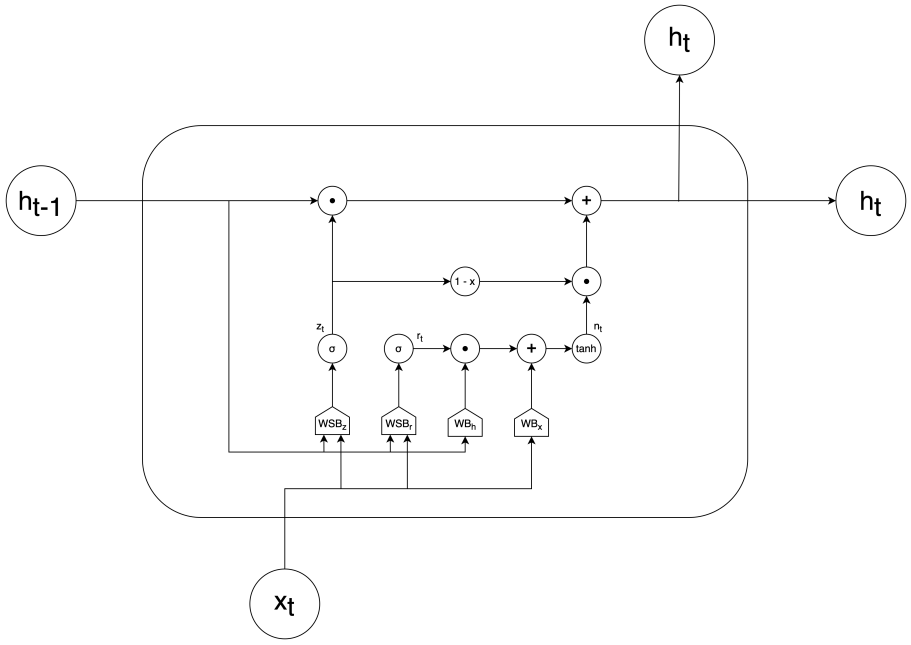


Figure 8: GRU Cell

2.2.4 Optuna

Optuna is optimization software for hyperparameter tuning, created by Akiba et al. [9]. In addition to searching the parameter space, the software includes a pruning feature and a dashboard for monitoring the training phase. The user defines the search space and the scaling method for each parameter. Optuna uses the following techniques to optimize the parameters:

The sampling method of Optuna is based on Tree Parzen Estimator (TPE)[10] and Covariance Matrix Adaption Evolution Strategy (CMA-ES)[11]. TPE uses independent sampling of parameters, while CMA-ES uses the covariances of parameters to search for the optimal combination of parameters. The paper uses TPE for the first 40 steps; then, the CMA-ES method is used. TPE creates two distributions for the best and worst halves of a parameter based on the objective scores. The following parameter is sampled from where the worst distribution is minimized and the best distribution is maximized. CMA-ES is an evolution-based algorithm. The algorithm uses the covariances of parameters to move the sampling space towards the global optimum.

Asynchronous Successive Halving (ASHA) is used for pruning in Optuna. ASHA sequentially tests if the intermediate value of the current trial is in the top half of all intermediate values at the same time step. The asynchronous behavior makes it possible for multiple workers to prune trials without waiting for the information of other workers. Figure 9 shows the optuna dashboard, with a plot of each combination of hyperparameters and their corresponding objective value.

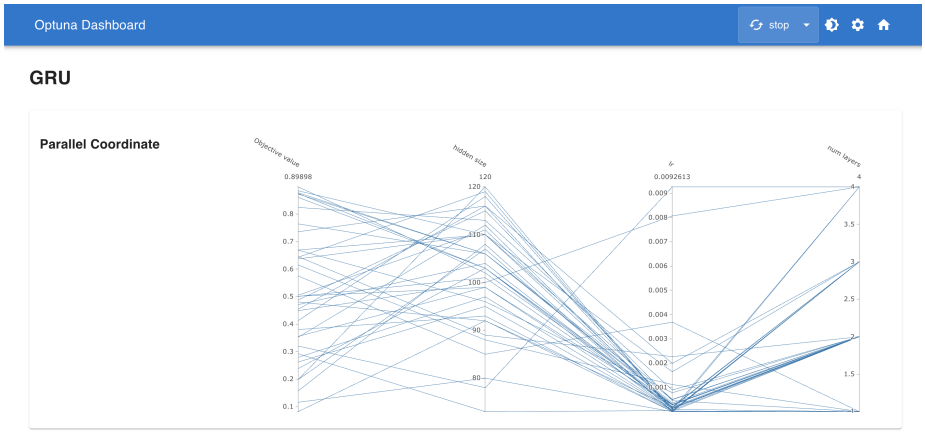


Figure 9: Optuna Dashboard

3 Related Work

This chapter is written as a Structured Literature Review (SLR). SLR is a formal way of synthesizing available information from primary studies [12] and is described in section 3.1. Studies related to the research questions are searched for and evaluated based on quality. This process is described in section 3.2. The results of the SLR method are a set of related studies of high quality. Extracted information from these studies is presented in section 3.3. The relevance of the studies relating to the research questions is discussed in section 3.4.

3.1 Structured Literature Review (SLR)

The review method was based on the method of Kofod-Petersen, which was outlined in *How to do a Structured Literature Review in computer science* [12]. The main guidelines of the SLR method were used, and only minor alterations were made. The quality screening criteria were identical to the set that KofodPetersen suggested. The inclusion criteria were specific for this thesis but based on his examples. In addition to the searching- and snowballing methods for finding studies, recommended studies were added.

3.1.1 Primary Studies

According to the SLR method of Kofod-Petersen, primary studies are found by doing the following:

1. Search the relevant subdomain
2. Select primary studies based on inclusion criteria
3. Filter primary studies based on quality assessment

The relevant subdomain of research for primary studies is the intersection of several domains. Kofod-Petersen used groups of synonymous, key terms to describe these domains. Interchangeable terms improve the method's robustness, since several studies might be related to precisely the same topic even though they are using different words. The goal should be to get a varied set of studies that can span the whole subdomain with snowballing. However, the attention should also be focused on the most relevant studies by narrowing the search query.

The key terms can be arranged in a table like Table 1, where the columns represent each domain and the rows represent different ways of expressing them by words. When searching for the relevant subdomain, OR-operators should be used within each group, and AND-operators should be used between each group.

Table 1: Synonyms for Each Group

	Group 1	Group 2	Group 3
Term 1	$Synonym_{1,1}$	$Synonym_{2,1}$	$Synonym_{3,1}$
Term 2	$Synonym_{1,2}$	$Synonym_{2,2}$	$Synonym_{3,2}$
Term 3		$Synonym_{2,3}$	

Figure 10 shows the interesection of the groups, which results in the relevant subdomain for primary studies.

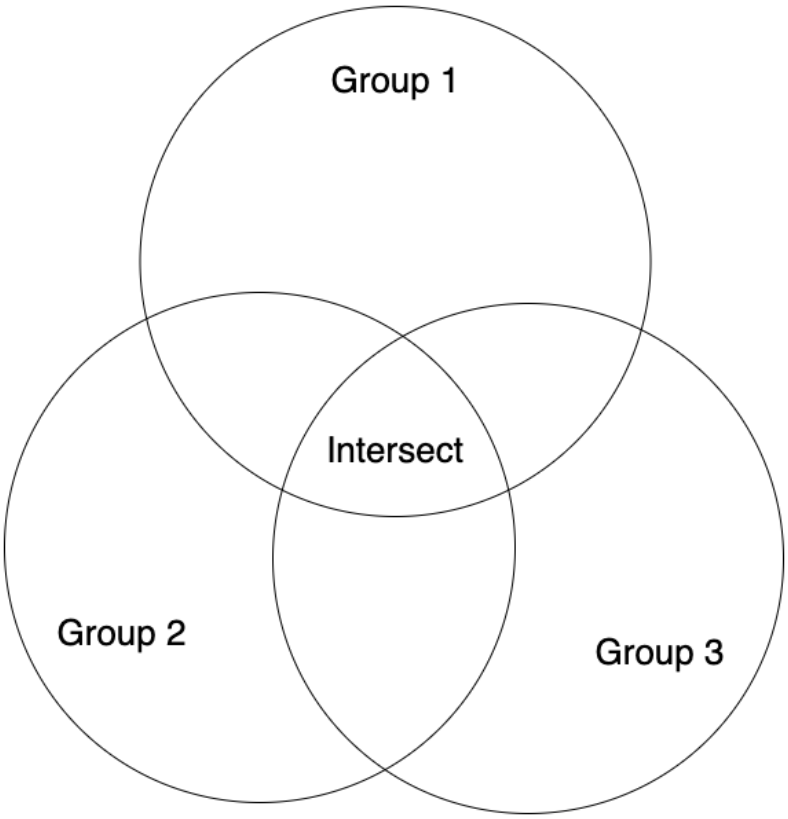


Figure 10: Intersect of Groups for Primary Studies

The search results will be too large to include in the quality assessment. Therefore the inclusion criteria must be set only to select the relevant studies. Kofod-Petersen stated that the inclusion criteria should be specifically tailored

to each problem, but some general principles apply regarding exclusion [12]:

- Duplicates (keep the highest-ranking source)
- The same study was published in different sources (keep the highest-ranking source)
- Studies are published before a specific date (or even after).

A study that was found using this approach is denoted as SQ.

3.1.2 Recommended Studies

The recommended studies constituted a varied set of challenges and solutions in musical style. They were, in different ways, linked to the research questions of this thesis. Supervisors of the research project recommended the studies. A study that was found using this approach is denoted as SR.

3.1.3 Secondary Studies

After the primary and recommended studies was been assessed, the snowballing method was used to find secondary studies. Snowballing refers to using the reference list or the citations of a study to identify additional studies [13]. Backward snowballing is done by exploring the references of a study to find previously written material on the same subject. Forward snowballing is to explore newly written material on a given topic by finding studies that cite a previously written study on the same issue. Even though Wohlin outlined a great method of evaluating intermediary studies found by the snowballing method [13], it will not be used. All seemingly relevant studies was added to the set of secondary studies and evaluated based on the method of KofodPetersen [12]. This was done to assess the studies consistently. Both studies found directly and indirectly from the starting set constituted the set of secondary studies. A study that was found using this approach is denoted as SS.

3.1.4 Quality Assessment

The quality assessment is crucial in selecting relevant studies to base the research on. Only primary and recommended studies that have passed the quality assessment was used to find secondary studies. Kofod-Petersen categorized the quality assessment of studies into three screenings. Each of these stages has their own criteria and level of detail:

1. Primary Inclusion Criteria - Abstract Screening
2. Secondary Inclusion Criteria - Full-Text Screening
3. Quality Criteria - Full-Text Assessment

The primary and secondary screenings filter studies that are not thematically relevant. The quality assessment stage is needed to ensure a high quality of the selected primary studies. Table 2 contains the inclusion criteria that each study had to satisfy to be evaluated based on quality. IC 4 was the only screening criteria not linked to thematic relevance. It was added to filter studies that were difficult to access, even though they were easy to find. Table 3 shows the proposed quality screening criteria of Kofod-Petersen, which was also used in this thesis [12].

Table 2: Inclusion Criteria

ID	Screening	Criteria
IC 1	Primary	The study is in the domain of MIR
IC 2	Primary	The study is a primary study, presenting empirical results
IC 3	Primary	The study describes processing of music data
IC 4	Primary	The study is easily accessible
IC 5	Secondary	The study focuses on features which are relevant for piano
IC 6	Secondary	The study includes useful elements with regard to performance style

Table 3: Quality Criteria

ID	Criteria
QC 1	Is there is a clear statement of the aim of the research?
QC 2	Is the study is put into context of other studies and research?
QC 3	Are system or algorithmic design decisions justified?
QC 4	Is the test data set reproducible?
QC 5	Is the study algorithm reproducible?
QC 6	Is the experimental procedure throughly explained and reproducible?
QC 7	Is it clearly stated in the study which other algorithms the study's algorithm(s) have been compared with?
QC 8	Are the performance metrics used in the study explained and justified?
QC 9	Are the test results thoroughly analysed?
QC 10	Does the test evidence support the findings presented?

In the quality screening, the studies were evaluated if they satisfied (1 point), partly satisfied (0.5 points), or did not satisfy (0 points) each of the criteria. The aggregated score was compared to a particular threshold. All the studies with a score larger or equal to the threshold passed the quality assessment. Additionally, studies could not achieve 0 points for QC5. The method of the study must in some way be possible to reproduce.

3.2 Conducting SLR

This section describes the implementation of the SLR method. That includes the decisions that were made, which studies were found and how the studies performed in the quality assessment.

3.2.1 Primary Studies

The research questions were located at the intersection of the following domains: Artificial intelligence, music and style. The intersection of these domains corresponds to many subfields with a wide variety of data types and methods. Thus, the intersection needed to be reduced when searcing for related literature. Each domain will be described by a few key terms used in the search for primary studies.

This thesis focused on the challenges of music linked to piano performances. The main reason for using piano is that the instrument imposes a limited way of applying style for the performer. For string instruments, the performer can add vibrato, which is hard to quantify and use for research purposes. Also, the way the performer touches the strings is hard to quantify and will affect the style. In the piano example, the performer can deviate in timing, velocity and pedaling. These are easy to quantify and should be ideal as input data for machine learning models.

The focus of the thesis was also to look at style on a note level. Piano pieces recorded in MIDI format are ideal for working with since all the input information is easily extractable. However, other data types and approaches need to be considered to answer **RQ1**. Music was not used as a key term because it was considered to be too broad. Key terms: **piano, midi**

The research questions are linked to the individual styles of different piano performers. The results can vary significantly based on the composer and piece by looking at the performances themselves. Thus, the goal was to isolate the performers' styles from the pieces they are playing. This was derived from the deviations between the performer and the score. Also, the tools and methods of the performer that are not defined in the score could be used. A score is often described as robotic and lacks the human interpretation that makes the music enjoyable. The word expressive is typically used to describe the addition of style to a score to make it humanlike or enjoyable.

The approaches connected to emotion in music was considered appropriate for style. Emotion and style are closely related; both are high-level features of music. Even though the output is different, the underlying methods and approaches might pertain to both subdomains. Music Emotion Recognition (MER) is a highly researched field and could offer great insight into music and ways to handle it.

Key terms: **style, expressive, emotion**

When it comes to the technical challenges of the research questions, they are linked to the prediction of mid-level features (**RQ1**) and the recognition of

performers (**RQ2**). Solutions in this domain of challenges typically do both in a disentangled step. Emotions, styles of composers, or performers are directly classified from the music. These approaches could be altered to output a set of mid-level features, as long as there was data for training. Classification is a closely related synonym of recognition.

Key terms: **prediction, recognition, classification**

Kofod-Petersen presented the following resources in his outline of the SLR method: ACM digital library, IEEE Xplore, ISI web of knowledge, ScienceDirect, CiteSeer, SpringerLink and Wiley Inter Science [12]. Even though these resources were great for the domain of computer science, they did not include a lot of articles on music and performeric style. The International Society for Music Information Retrieval (ISMIR), on the other hand, published a lot of articles on this subject. In conjunction with their yearly conference, they posted all the accepted papers on their website. The advantage of using such resources was that the autenticity and validity of papers could be trusted. Yet, all the mentioned resources only contained subsets of the entire domain and had varied possibilities when it came to searching and filtering.

Searching and filtering were handled well by Google Scholar. The search engine included the possibility of using boolean expressions when constructing the search query. This functionality made it possible to combine all potential combinations of synonyms into a single query. Thus, the risk of not using all possible combinations was removed. Google Scholar could not verify the quality of all the articles, but it could redirect the user to external resources that the user could trust. Even though Google Scholar could not guarantee to find all or the most relevant articles, it was considered one of the best at both of these challenges. Google Scholar was therefore chosen as the searching resource.

In order to search for primary studies relating to the research questions, the key terms were used. Table 4 shows the SLR representation of the terms and their corresponding group.

Table 4: Search terms for SLR method

	Group 1	Group 2	Group 3
Term 1	piano	style	classification
Term 2	midi	expressive	prediction
Term 3		emotion	recognition

The search expression inserted into Google Scholar was: (piano OR midi) AND (style OR expressive OR emotion) AND (classification OR prediction). The principles described in subsection 3.1.1 were used, and the year limit was set from 2015 to 2021. Since the literature review was completed in the beginning of 2022, the end limit was set to 2021 for reproducibility. The start limit was set to 2015 in order to get recent and relevant research. The search query gave 23300 results (0.06 sec). The first two pages of results were selected for further screenings, 20 studies in total. These are shown in Table 5.

Table 5: Studies from Search Query

ID	Title	Authors	Year
SQ1	EMOPIA: A Multi-Modal Pop Piano Dataset For Emotion Recognition and Emotion-based Music Generation	Hsiao-Tzu Hung, Joann Ching, Seungheon Doh, Nabin Kim, Juhan Nam, Yi-Hsuan Yang	2021
SQ2	Composer Style Classification of Piano Sheet Music Images Using Language Model Pretraining	TJ Tsai, Kevin Ji	2020
SQ3	A Scheme of MIDI Music Emotion Classification Based on Fuzzy Theme Extraction and Neural Network	Peilin Chen, Lei Zhao, Zongyu Xin, Yumeng Qiang, Ming Zhang, Tiemeng Li	2016
SQ4	An Analysis of Low-Arousal Piano Music Ratings to Uncover What Makes Calm and Sad Music So Difficult to Distinguish in Music Emotion Recognition	Yu Hong, Chuck-Jee Chau, Andrew Horner,	2017
SQ5	Recurrent Neural Network for MIDI Music Emotion Classification	Wei Zhao, Yinan Zhou, Yun Tie, Yushu Zhao	2018
SQ6	Composer Recognition based on 2D-Filtered Piano-Rolls	Gissel Velarde, Tillman Weyde, Carlos Cancino Chacón, David Meredith, Maarten Grachten	2016
SQ7	Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions	Yu-Siang Huang, Yi-Hsuan Yang	2020
SQ8	From Content-based Music Emotion Recognition to Emotion Maps of Musical Pieces	Jacek Grekow	2018
SQ9	Musical instrument emotion recognition using deep recurrent neural network	Sangeetha Rajesh, NJ Nalini	2020
SQ10	Music recognition without identification and its relation to déjà entendu: A study using “Piano Puzzlers”	Katherine L. McNeely-White, Anne M. Cleary	2019
SQ11	MIDI-VAE: Modeling Dynamics and Instrumentation of Music with Applications to Style Transfer	Gino Brunner, Andres Konrad, Yuyi Wang, Roger Wattenhofer	2018
SQ12	Measurement, recognition, and visualization of piano pedaling gestures and techniques	Beici Liang, György Fazekas, Mark Sandler	2018
SQ13	A Multimodal Music Emotion Classification Method Based on Multifeature Combined Network Classifier	Changfeng Chen, Qiang Li	2020
SQ14	Automated composer recognition for multi-voice piano compositions using rhythmic features, n-grams and modified cortical algorithms	Nadine Hajj, Maurice Filo, Mariette Awad	2018

SQ15	Tonal complexity features for style classification of classical music	Christof Weiss, Meinard Müller	2015
SQ16	Music to my ears: Age-related decline in musical and facial emotion recognition.	Ryan Sutcliffe, Peter G. Rendell, Julie D. Henry, Phoebe E. Bailey, Ted Ruffman,	2017
SQ17	Piano Pedaller: A Measurement System for Classification and Visualisation of Piano Pedalling Techniques	Beici Liang, György Fazekas, Andrew McPherson, Mark Sandler	2017
SQ18	Music emotion recognition using chord progressions	Yong-Hun Cho, Hyunki Lim, Dae-Won Kim, In-Kwon Lee	2016
SQ19	A Computational Study of the Role of Tonal Tension in Expressive Piano Performance	Carlos Cancino-Chacón, Maarten Grachten	2018
SQ20	Analysis of the Practical Value of Czerny Piano Etudes	Aiyong Xin	2018

3.2.2 Recommeded Studies

The recommended studies were provided by Jong Hwa Park, professor at the College of Music at Seoul National University. Table 6 shows the recommended studies.

Table 6: List of Recommended Studies

ID	Title	Authors	Year
SR1	A data-driven approach to mid-level perceptual musical feature modeling	Anna Aljanaki, Mohammad Soleymani	2018
SR2	Towards Explaining Expressive Qualities in Piano Recordings: Transfer of Explanatory Features Via Acoustic Domain Adaptation	Shreyan Chowdhury, Gerhard Widmer	2021
SR3	VirtuosoNet: A Hierarchical RNN-based System for Modeling Expressive Piano Performance	Dasaem Jeong, Taegyun Kwon, Yoojin Kim, Kyogu Lee, Juhan Nam	2019
SR4	Performance Error Detection and Post-Processing for Fast and Accurate Symbolic Music Alignment	Eita Nakamura, Kazuyoshi Yoshii, Haruhiro Katayose	2017

3.2.3 Secondary Studies

Secondary studies were found using the snowballing method (subsection 3.1.3). Table 7 shows the results. The *SID* column conveys the ID of the study which lead to the given study. The *Method* column displays the snowballing method that was used to find the given study.

Table 7: Secondary Studies

ID	Title	Authors	Year	SID	Method
SS1	Detecting Emotions in Classical Music from MIDI Files	Jacek Grekow, Zbigniew W. Ras	2015	SQ1	Backward
SS2	Modeling Musical Style with Language Models for Composer Recognition	María Hontanilla, Carlos Pérez-Sancho, Jose M. Iñesta	2013	SQ6	Backward
SS3	On the Characterization of Expressive Performance in Classical Music: First Results of the Con Espressione Game	Carlos Cancino-Chacón, Silvan Peter, Shreyan Chowdhury, Anna Aljanaki, Gerhard Widmer	2020	SR2	Backward
SS4	Computational Models of Expressive Music Performance: A Comprehensive and Critical Review	Carlos E. Cancino-Chacón, Maarten Grachten, Werner Goebl, Gerhard Widmer	2018	SS3	Backward
SS5	Using listener-based perceptual features as intermediate representations in music information retrieval	Anders Friberg, Erwin Schoonderwaldt, Anton Hedblad, Marco Fabiani, Anders Elowsson	2014	SS3	Backward
SS6	A formalization of relative local tempo variations in collections of performances	Jeroen Peperkamp, Klaus Hildebrandt, Cynthia Cheng Sien Liem	2017	SS4	Backward
SS7	Expressive timing from cross-performance and audio-based alignment patterns: an extended case study	Cynthia C.S. Liem, Alan Hanjalic	2011	SS4	Backward
SS8	A phylogenetic approach to music performance analysis.	Elad Liebman, Eitan Ornoy, Benny Chor	2012	SS4	Backward
SS9	Towards computer-assisted understanding of dynamics in symphonic music	Maarten Grachten, Carlos Eduardo Cancino-Chacón, Thassilo Gadermaier, Gerhard Widmer	2017	SS4	Backward
SS10	An evaluation of linear and non-linear models of expressive dynamics in classical piano and symphonic music	Carlos Eduardo Cancino-Chacón, Thassilo Gadermaier, Gerhard Widmer, Maarten Grachten	2017	SS4	Backward

SS11	MidiBERT-Piano: Large-scale Pre-training for Symbolic Music Understanding	Yi-Hui Chou, I-Chun Chen, Chin-Jui Chang, Joann Ching, Yi-Hsuan Yang	2021	SQ1	Forward
SS12	Performer Identification From Symbolic Representation of Music Using Statistical Models	Syed Rifat Mahmud Rafee, Gyorgy Fazekas, Geraint A.Wiggins	2021	SQ12	Forward
SS13	Towards Explainable Music Emotion Recognition: The Route via Mid-level Features	Shreyan Chowdhury, Andreu Vall, Verena Haunschmid, Gerhard Widmer	2019	SR1	Forward
SS14	On Perceived Emotion in Expressive Piano Performance: Further Experimental Evidence for the Relevance of Mid-level Perceptual Features	Shreyan Chowdhury, Gerhard Widmer	2021	SR1	Forward
SS15	Tracing Back Music Emotion Predictions to Sound Sources and Intuitive Perceptual Qualities	Shreyan Chowdhury, Verena Praher, Gerhard Widmer	2021	SR2	Forward
SS16	An Interdisciplinary Review of Music Performance Analysis	Alexander Lerch, Claire Arthur, Ashis Pati, Siddharth Gururani	2021	SR3	Forward
SS17	Computational Analysis and Modeling of Expressive Timing in Chopin Mazurkas	Zhengshan Shi	2021	SR3	Forward
SS18	Score and Performance Features for Rendering Expressive Music Performances	Dasaem Jeong, Taegyun Kwon, Yoojin Kim, Juhan Nam	2019	SR4	Forward
SS19	Rendering Music Performance With Interpretation Variations Using Conditional Variational RNN	Akira Maezawa, Kazuhiko Yamamoto, Takuya Fujishima	2019	SS4	Forward
SS20	Modeling and Estimating Local Tempo: A Case Study on Chopin's Mazurkas	Hendrik Schreiber, Frank Zalkow, Meinard Müller	2020	SS6	Forward

3.2.4 Quality Assessment

All types of studies (SQ, SR and SS) were evaluated based on the quality assessment described in subsection 3.1.4. When a study did not pass the quality assessment, the screening criteria column states the reason for failure. For the studies that passed the primary and secondary screenings, their quality scores are shown in the last column. Only the studies with quality scores of 7 and higher passed the quality screening. SQ3, SQ5, SQ17 did not satisfy the score

requirement. SQ4, SQ10, SQ16 and SQ20 did not include processing of musical data. SQ8 was only attainable by purchase. SQ18 focused on the effect of chord progression for emotion recognition. This relates only to the style of the composer and was not applicable to performance style. The results for the primary studies are shown in Table 8. The quality scores of all the studies are presented in the appendix (A1).

Table 8: Results of Quality Assessment of Primary Studies

ID	Pass	Screening Criteria	Quality Score
SQ1	Yes		8
SQ2	Yes		9,5
SQ3	No	Quality Score	5
SQ4	No	IC 3	-
SQ5	No	Quality Score	6
SQ6	Yes		9
SQ7	Yes		8
SQ8	No	IC 4	-
SQ9	Yes		7,5
SQ10	No	IC 3	-
SQ11	Yes		9,5
SQ12	Yes		7,5
SQ13	Yes		8,5
SQ14	Yes		8,5
SQ15	Yes		7
SQ16	No	IC 3	-
SQ17	No	Quality Score	5,5
SQ18	No	IC 6	-
SQ19	Yes		8,5
SQ20	No	IC 3	-

All of the recommended studies passed the quality assessment, and Table 9 shows the quality scores.

Table 9: Results of Quality Assessment of Recommended Studies

ID	Pass	Screening Criteria	Quality Score
SR1	Yes		8
SR2	Yes		9,5
SR3	Yes		8,5
SR4	Yes		10

Table 10 shows the result of the quality assessment of the secondary studies. SS4 and SS16 are both reviews of the domain, and did not present any new methods or results. However, they were great for high level overview of the

existing approaches and further snowballing. SS5 is a book which was only attainable by purchase. SS8 focused on features and methods regarding the violin, which were not applicable for piano. SS2 did not mention the language model that was used or explain the method thoroughly. Thus, the study scored 0 points for QC5 and did not pass the quality assessment. SS1, SS6 and SS18 did not satisfy the quality score requirement of 7.

Table 10: Results of Quality Assessment of Secondary Studies

ID	Primary Study	Screening Criteria	Quality Score
SS1	No	Quality Score	5
SS2	No	QC5	-
SS3	Yes		8,5
SS4	No	IC2	-
SS5	No	IC4	-
SS6	No	Quality Score	6,5
SS7	Yes		9
SS8	No	IC5	-
SS9	Yes		9
SS10	Yes		8
SS11	Yes		8,5
SS12	Yes		7
SS13	Yes		9
SS14	Yes		8,5
SS15	Yes		7,5
SS16	No	IC2	-
SS17	Yes		7,5
SS18	No	Quality Score	3
SS19	Yes		7,5
SS20	Yes		9

3.3 Results of SLR

The primary and secondary studies represented various subdomains linked to musical style. Some of the studies compared music with different compositions. These comparisons primarily highlighted the composition style of the music, not the expressive style. This class of studies is denoted as *Inter-Composition*. However, studies that focused on expressive style within a specific composition are labeled as *Intra-Composition*. The studies related to *Intra-Composition* and *Inter-Composition* are presented in subsection 3.3.1 and subsection 3.3.2, respectively. The relevance of the studies concerning the research topic is discussed in subsection 3.4.1.

3.3.1 Intra-Composition

Jeong et al. combined a Hierarchical Attention Network (HAN) and a Conditional Variational Autoencoder (CVAE) to extract performance style and generate expressive performances (SR3)[14]. The score encoder consisted of a note encoder using LSTM and beat/measure extraction methods using attention. Extracted information was given to the CVAE for probabilistic autoencoding. The latent vector was used to extract style and transfer it to another piece. Piano performances were collected and transcriptions were manually refined for consistency. Two versions were tested, with measure-level modules (HAN-M) or without (HAN-S). The method was validated with a listening test. The authors stated that their proposed models (HAN-M, HAN-S) perform better than Basis Mixer (BM) and a baseline LSTM in 7 different categories. The model was praised for its human expressiveness by listeners, but it also got some negative feedback. The authors stated that the predicted pedal usage was often too deep and dirty, or too shallow.

Nakamura et al. proposed a method for aligning polyphonic performances in symbolic format to the score or other performances (SR4)[15]. The method handled wrongly played notes, which could be extra notes, missing notes or wrong pitch. MIDI files were used for audio representation and merged-output Hidden Markov Models (HMM) for realignment. The method was compared with several methods on three datasets. It performed better in terms of accuracy and efficiency on all tests.

Liang et al. used an optical sensor and piano recordings to create a dataset for sustain pedal usage (SQ12)[16]. Signal processing was used to determine the ground truth labels for part-pedaling, onset- and offset timings. Support Vector Machine (SVM), HMM, k-Nearest Neighbors (k-NN), Decision Tree (DT), Random Forest (RF) and Gaussian Naive Bayes (GNB) were compared in the classification task. The SVM method performed best. It was able to distinguish the quarterly levels of part-pedaling.

Cancino-Chacón and Grachten used recurrent neural networks and measures of tonal tension to predict expressive performances of Mozart's sonatas (SQ19)[17]. Three quantities were used to capture the tonal tension, first proposed by Chew[18]: cloud diameter, cloud momentum and tensile strain. The

extraction method of Herremans and Chew[19] was used to capture these quantities. The authors concluded that the method could be used for relative tempo and dynamics, but only when low level pitch information was available. The dependence on this information makes the tension quantities unable to capture the expresiveness fully.

Cancino-Chacón et al. did an online questionaire to explore the expressive character of classical piano performances. (SS3)[20]. The participants listened to 9 classical pieces by different performers and were asked to describe the performances in free text and choose their favorite. The resulting dataset was tested for semantic simmilarity, performance preference and PCA word dimension correlation with mid-level features. The authors stated that different performances of the same piece tended to be described in a similar manner. Three of the pieces rejected the null hypothesis of not a preferred performance with an alpha of 0.01, and five at the 0.05 level. The authors showed that there are relationships between the mid-level features and the word dimensions: "For instance, the analysis suggests that louder performances or performances with large outliers in the valence curve would be perceived as more irregular and agitated."

Liem and Hanjalic used an entropy based measuring method to study alignment differences between performances (SS7)[21]. The method was based on Chroma Discrete Cosine Transform-Reduced Log Pitch (CRP) and Dynamic Time Warping (DTW), which outputs aligned local tempos. The authors stated that the entropy based method captured only the diversity of local errors. This is in contrast to standard deviation, which captured the aggreagation local errors. The authors ran two tests to verify that both standard deviation and entropy were not returning sequences of random noise. They concluded that the methods were strongly correlated, yet using entropy entailed less noise and amplification of positive peaks.

Grachten et al. used basis functions that were extracted from MusicXML files to predict dynamics (SS9)[22]. The method was usable for both single instruments and orchestras. When there were several instruments, all the basis functions of all the intruments were merged together in a specific way. The authors compared several model architectures: linear, bidirectional RNN and Feed Forward Neural Network (FFNN). The bidirectional RNN performed the best, with the FFNN close behind. However, the authors stated that there could have been a possibility of overfitting for both methods, because of the small dataset.

Cancino-Chacón et al. built upon the work regarding basis functions by thorougly testing non linear models on several datasets (SS10)[23]. The datasets comprised of large amounts of notes, but only a few players and composers: Magaloff/Chopin, Zeilinger/Beethoven, RCO/Symphonic. The authors used the same methods as Grachten et al.[22], in addition to a model with the LSTM architecture. A method for calculating the contribution of each basis function to the total variance of the model was also proposed. The authors concluded that the non linear models were substantially more accurate and also able to learn interaction effects of the score. Additionally, the analysis of the non linear models indicated the validity of previous hypotheses such as high notes should

be played louder.

Rafee et al. aligned each performance with the average performance of the same piece and use the deviation distributions to classify the performer (SS12)[24]. The features were extracted from the MIDI files: Onset time, inter onset interval, offset time duration, dynamic level and note duration. All the deviation distributions were modeled by histogram, KDE and Gaussian Mixture Model (GMM). The KL-divergence was computed for all combinations of the unknown performer and a known performer. The authors used equal weighting when combining the KL-divergence of each feature. The unknown performer was classified as the performer with the smallest amount of divergence. Using the average performance as a reference point worked well according to the authors, as long as there were enough performances for the same piece. The combination of inter inset interval, dynamic level and note duration performed the best for histogram and KDE. Onset time performed the best for single features.

Shi used a recurrent neural network to predict expressive piano timings in Chopin Mazurkas (SS17)[25]. The proposed method preprocessed the data based on the inter-beat-interval of the score and a chosen value for the mean tempo. A bidirectional LSTM was used to learn signature rhythms of piano performers. Performances were generated based on the score of different pieces, then compared to human performances and generated performances by Virtuosonet [14]. Shi showed that the proposed method had a higher correlation of relative tempo to human performances than Virtuosonet. Furthermore, the method was better at capturing the rhythmic structures of Chopin Mazurkas.

Maezawa et al. proposed a Conditional Variatonal Recurrent Neural Network (CVRNN) to extract performance style from piano recordings and generate new performances (SS19)[26]. The generative model was decoupled from the score in order to learn piece independent variability in performances. The proposed method was compared to the Finale software, human playing and score recreation in a listening test. The model produced better results in naturalness and worse in expresiveness when compared to the human in the listening test. Pearson's correlation coefficient was used to qualitatively test the model's ability to predict the next notes of a human performance. Based on the results, the authors concluded that the model was able to recreate the style of a performer when given a snippet.

Schreiber et al. trained Convolutional Neural Nets (CNNs) to model local tempo and measure tempo stability (SS20)[27]. The models were based on the DT-Maz architecture, and were compared to the DeepTemp- and Böck-software. Two models were trained, with different splits of the dataset: Each fold included all versions for the same piece, or all pieces for the same version. Local tempo was modeled by median aggregated inter beat interval, and tempo stability by the coefficient of variations of local tempo values. The models were specifically trained in the piano domain, and performed better than the general purpose softwares on the modelling task. The differences in training splits were used to learn differences of the Mazurka pieces.

3.3.2 Inter-Composition

Aljanaki and Soleymani collected a dataset annotated with 7 mid-level features, and used mel-spectogram and CNN to recognize emotions (SR1)[1]. The participants did pairwise comparison to create absolute scales for each dimension, then annotated each song according to the scales. Only users who passed a musical test were recruited for the annotation. If they answered randomly or far away from the average, their annotations were not used in the final dataset. The authors concluded that the mid-level features can model different emotions well, despite having no information about tempo or loudness. The annotation consistencies for the features were high, except for rhythmic tonal stability and complexity.

Chowdhury and Widmer presented a three-step approach to adapt a mid-level feature extraction method to the domain of expressive piano performances (SR2)[28]. The extraction method of Aljanaki and Soleymani [1] was based on songs from different wide variety of genres. Thus, the MAESTRO dataset was added to make a distilled student model adaptive to the piano domain. Different model architectures were tested: VGG, RF-ResNET, RF-ResNet with domain adaption and RF-ResNet with domain adaption and teacher student learning. The authors concluded that the performance was significantly improved by the domain adaption and the teacher student training scheme. They also believed the method showed potential for further adaptions in the realm of musical style.

Hung et al. collected a multimodal dataset of pop piano music, with audio- and MIDI-representation, for tasks related to emotions (SQ1)[29]. The MIDI files were automatically transcribed from the audio of downloaded Youtube videos. The dataset contained only solo performances of professional piano players which conveyed a specific emotion. Every piece was assigned to a specific quadrant in the valence-arousal plane. Emotion recognition with symbolic data was tested with a LSTM model, and music in Revamped MIDI Derived Events- (REMI) and MIDI-format. With audio data, CNN was compared to the combination of MFCC features and logistic regression. Symbolic domain classifiers performed better with regard to valence. Yet, no model outperformed the others in all areas. Emotion generated music was also tested with Compound Word Representation (CP) and a transformer model. The authors concluded that it was possible to generate music, given a specific target emotion, to a certain degree.

Tsai and Ji used Natural Language Processing (NLP) with unlabelled pretraining to classify the composer of a sheet of music (SQ2)[30]. The sheets were converted into sequences of words and used to train the language model in an unsupervised manner. A text classifier was added to classify composers. The language models used are AWD-LSTM, GPT-2 and RoBERTa. In addition, these architectures were compared with a CNN model, with a linear classifier instead of the text classifier. The transformer models (GPT-2, RoBERTa) performed better on the full page classification task. The pretraining improved the performance significantly. The tSNE-plot of music by five different composers showed clear clustering.

Velarde et al. used a CNN, pitch-time representations (piano-rolls) and filters to classify composers from MIDI files (SQ6)[31]. The method used no form of feature extraction or splitting the input into several voices. A comparison was conducted of Chromatic pitch and Morphetic pitch for preprocessing, and of Morelet Wavelet and Gaussian for filters. Different input lengths and centering methods were also compared. The best combination of settings achieved state-of-the-art performances on the Haydn/Mozart discrimination task. The authors show that the filtering significantly improved the performance of the method.

Huang and Yang proposed the REMI representation for audio data and tested its appropriateness with a music generation task (SQ7)[32]. The method was supposed to represent the bar structure of music better than MIDI. This is beacuse REMI used note positions (in accordance to a bar) to describe the position of a note instead of time shifts. The proposed method also used duration instead of note-off events, to easily capture the dependence of beginning and end of notes. In addition, REMI had local tempo changes as a parameter and chords as a seperate musical token. Several transformers were trained with different data representations and their generated music was evaluated in a listening test. The REMI representation achieved the highest scores, with large margins, both for the professional and the amateur group. Objective evaluation of the models indicated that REMI produced music with less deviations in rhythm.

Rajesh and Nalini compared different feature extraction methods and models, to classify instruments and emotions (SQ9)[33]. The extraction methods include Mel Frequency Cepstral Coefficient (MFCC), Chroma Energy Normalized Statistics (CENS) and Chroma Short-Time Fourier Transform (STFT). SVM and LSTM was used for classification. The combination of MFCC and LSTM performed the best, with a significant margin. It should be noted that the MFCC method extracted the largest set of features in the study, more than the rest of the methods combined. The authors concluded that the instrument impacts the emotion classification of monophonic instrument recordings.

Brunner et al. used a Variational Autoencoder (VAE) to apply style transfer to polyphonic music (SQ11)[34]. The autoencoder handled three kinds of data features in the form of rolls: Pitch, instrument and velocity. The model was comprised of dense layers and GRU. It was trained on a dataset including the genres: Jazz, Pop and Classic. Style classifiers for each data feature were trained to test the model's ability to successfully transfer style. The style transfer affected the classification, but only slightly. When changing from jazz to pop, all the instruments changed. Latent space evaluation with tSNE-plots showed clusters and a disentangled dimension from jazz to classic along the x-axis.

Chen and Li proposed a multimodal model for classifying emotions (SQ13)[35]. Both lyrics and audio were processed in several dimensions. Chi-squared method was used to extract one dimensional information from the lyrics. Feature extraction methods, such as MFCC and Zero Crossing Rate (ZCR), were used to do the same for audio. An architecture with CNN and LSTM was used to model two dimensional input of audio and lyrics. Spectograms and Word2vec were used for this embedding. Both dimensions of each modal were merged into one classifier. Then, a softmax layer was added to merge the two modals into

one classifier. The multimodal approach to emotion recognition produced better results than unimodal. The model of Chen and Li performed comparingly to other state-of-the-art methods for multimodal emotion recognition. However, it did not achieve the highest average accuracy.

Hajj et al. used n-grams and CA to recognize classical music composers (SQ14)[36]. The method distinguished voices in a MIDI score, constructed n-grams and generated a large feature vector. The vector was used to classify the composers with different methods: Cortial Algorithm (CA) was compared to SVM and one nearest neighbor classifier. CA performed the best. In addition, they were able to reduce the feature set to 0.1% and still achieve a high recognition rate.

Weiss and Müller proposed a new set of audio features which was tested on a classification task of historical music periods (SQ15)[37]. The features were based on chroma features. Four chroma extration methods were used, with four different time frames of smoothing. Half of the dataset consisted of orchestra music and the other half consisted of piano recordings. SVM and GMM were used for classification. The Linear Discriminant Analysis (LDA) plot showed clear clusterings of the historical periods. The authors stated that the results indicated the features ability to capture tonal aspects that are not related to timbre. The features also exhibited robustness towards artist- and composer-specific properties.

Chou et al. tested BERT's ability to tackle note level prediction, melody extraction, velocity prediction, composer classification and emotion classification (SS11)[38]. The input data was tokenized, both in REMI format and CP format. A baseline bidirectional LSTM model was used for comparison. Several datasets were used for different purposes: Pop17K and ASAP for pre-training, Pop909 for melody- and velocity prediction, Pianist8 for composer classification and EMOPIA for emotion classification. The study indicated that the CP representation performed better than REMI. MidiBERT performed well on several of the tasks, and beat the LSTM model in all of them. The authors also concluded that the pretraining of the models improved the performance significantly.

Chowdhury et al. explored the cost of using mid-level features as an intermediary step when classifying emotions (SS13)[2]. The first model consisted of two models that were independently trained: From audio to mid-level features and mid-level features to emotion (A2mid2E). The second model used both models, but was trained jointly (Joint). The last model classified emotion directly from the audio (A2E). The audio extraction part of the model consisted of CNN-layers and used spectrograms as input. Joint and A2mid2E performed slightly worse on the emotion classification task, 1% and 5% respectively. The joint method performed comparingly on the mid-level classification task. It can therefore be concluded that the method for jointly training optimized explainability and performance.

Chowdhury and Widmer tested several feature sets by their ability to predict emotions in the valence-arousal spectrum (SS14)[39]. The feature sets included: mid-level perceptual features, pre-trained emotion features, low level audio features, and score-based features. To test the methods, the authors collected a

dataset of ratings by 29 students. The average rating for each piece and piano performer was used as ground truth labels. Statistical methods, such as R^2 and correlation, were used to compare the feature sets. The authors concluded that the mid-level features are the most robust method. In addition to the ability of prediction, the features included intuitive musical meanings which humans could understand easily.

Chowdhury et al. used audioLIME and mid-level perceptuala features to explain models for emotion recognition (SS15)[40]. The study included three datasets, with different distributions of genres: Mid-level Perceptual Features Dataset, Database for Emotional Analysisin Music (DEAM) and Popular Music with Emotional Annotation (PMEmo). Performance of explainability was measured by complexity and fidelity. The authors stated that the metrics look related on a dataset level, but showed that it was not the case on a local level. The explainability was tested qualitatively by finding the cause of valence error for hip hop songs. This was traced back to the rhythmic stability of vocals. The authors explained that the model with valence error placed more importance on rhythmic stability than the one that was trained on multiple datasets. Thus, the explanation was considered to be correct.

3.4 Discussion

This section provides a discussion about the implementations and results of the SLR method. The relevance of the studies related to *Intra-Composition* and *Inter-Composition* are presented in subsection 3.4.1 and subsection 3.4.2, respectively. Subsection 3.4.3 discusses the choice of search terms. The consequences of selecting the given search terms is discussed in subsection 3.4.4.

3.4.1 Relevance of Intra-Composition

The method of Jeong et al. (SR3)[14] is related to the both research questions (**RQ1**, **RQ2**). Their score encoder module was a new approach of capturing note level interactions, which is important for mid-level features. The encoder part of the CVAE can be used to classify the style of performers.

The alignment method of Nakamura et al. (SR4)[15] is related to the extraction of mid-level features (**RQ1**). Alignment of performances is important in order to extract the deviations of each performer. The method is especially useful if the performances should be aligned to the score or directly to another performance. The study did not discuss the task of aligning a performance to the average performance of a given piece. This is possible with the proposed method as long as another method is used to compute the average performance.

The part-pedaling classification of Liang et al. (SQ12)[16] could be used in relation to mid-level features extraction (**RQ1**). The amount of sustain pedal usage is one aspect of performance style. To extract mid-level features from this information, its sequences and variations must be captured.

Since the musical style of an performer can be determined in a small time frame, the relative tempo and dynamics features of Cancino-Chacón and Grachten

(SQ19)[17] are relevant. The dependence on low level pitch information entails that there is a need for more features to distinguish expressive performances of piano performers (**RQ1**).

The dataset creation of Cancino-Chacón et al.(SS3)[20] entails several interesting discussions when it comes to participant knowledge and mid-level features. Their finding that the same pieces tend to be described simmilarly is a problem for the study of performance style. The style characteristics of an performer should ideally be independent of the piece (as long as that is the purpose of the performer), in order to be used for performer classification (**RQ2**). This suggests that the PCA method is not precise enough or that the set of performers are too similar with regard to style.

The measuring method of Liem and Hanjalic (SS7)[21] is related to the pre-processing of audio data. The feature was highly correlated with similar methods using standard deviation, yet it has some interesting properties. When preparing the audio data for mid-level features extraction (**RQ1**), several methods like this need to be tested. The proposed method might be the best way of processing the raw audio data for the models to understand the underlying relationships.

The dynamics that Grachten et al.(SS9)[22] predicted with their model is an important element of performance style. Even though the dynamics are noted in the score, there is no ground truth to what the correct loudness should be. In addition, piano performers can deviate from the timings of the score to express their own styles. The study's problem with lack of data is fixed in the work of Cancino-Chacón et al.(SS10)[23]. In this study, the authors provided larger datasets and more evidence. The prediction of loudness does not provide additional knowledge, since the information is already known from the velocity attribute in MIDI files.

The work of Rafee et al.(SS12)[24] is closely related to both research questions. The authors extract 5 low level features in order to classify piano performers (**RQ2**). Extracted low level features could be combined or used as input to a model in order to get mid-level features (**RQ1**). The study's use of distributions entailed the incosideration of note context. This is in contrast to mid-level features, which are mostly centered around the interaction of notes. Thus, it will be difficult to reuse methods like histogram, KDE and GMM.

The method of Shi (SS17)[25] predicted the timing variations of piano performers. This is an important feature when analyzing performance style. The proposed preprocessing is a great approach which might provide features that produce better results. Yet, the model for prediction is hard to reuse. The tempo variations could be extracted directly from the MIDI file, so the prediction provides no additional information.

The model of Maezawa et al.(SS19)[26] was able to distinguish performance styles (**RQ2**). Their feature extraction was extensive and combines several methods to optimize performance. The idea of using an ensemble model that handles different aspects of the input differently is good.

The measurement methods of Schreiber et al. (SS20)[27] are related to the mid-level features extraction (**RQ1**). These can be used to model the local

tempo variations of performers, which is an important aspect of performance style. The values can be used directly or in order to extract higher level concepts.

3.4.2 Relevance of Inter-Composition

The dataset collection and methods of Aljanaki and Soleymani (SR1)[1] are closely related to **RQ1**. They extracted mid-level features and used them to recognize emotion. However, the mid-level features used in this study might not produce the best results when it comes to performance style. The features are tightly coupled to the composition of a song, which can be more useful for emotion recognition. This is the same for the CNN based extraction method. It captured the features of the overall song, not the performance variations that are piece independent.

The adaption method of Chwodhury and Widmer (SR2)[28] is related to the research topic. The extraction method produced mid-level features and was adapted to the domain of performance style. This is exactly the domain of the research topic. However, the method introduces a lot of complexity through its three-step process. Using several techniques to adapt a model from another domain might produce worse results than creating a model specifically for the task domain.

The dataset and emotion tests of Hung et al. (SQ1)[29] are related to the extraction of mid-level features (**RQ1**). The study included a range of methods related to the audio format and their performances. The study focused on emotion recognition, where the differences in audio format is shown to not affect the performance significantly. This might not be the case in the domain of performance style.

The method of Tsai and Ji (SQ2)[30] is partly connected to the research topic. The model was difficult to use because of its goal of classifying composer style. In this thesis the goal of the method is to be independent of composer style and only capture the variations of the performers. However, the use of language models is an interesting approach which can be used to model the sequence of variations of performers.

The model of Velarde et al. (SQ6)[31] is connected to the research topic. The same model could be used to recognize performers of the same piece (**RQ1**, **RQ2**). However, the method is likely to be dependent on the piece. It might be possible to preprocess the data for independence and use the same method.

The data representation of Huang and Yang (SQ7)[32] is strongly related to the mid-level feature extraction (**RQ1**). The proposed method represented the information in relation to the basic structure of music, such that machines would have an implicit understanding of musical structure. This was specifically constructed for the domain of music generation. However, in the domain of performance styles, the underlying principles of deviations from the score or other performances must be represented. Thus, the data must be processed further in order to extract mid-level features, regardless of the input format.

The comparison of Rajesh and Nalini (SQ9)[33] highlighted several well known feature extraction methods related to audio recordings. The method

of representing music in spectrogram format can be difficult to use in the domain of performance style. This type of embedding is highly dependent on the piece and does not highlight the variations of individual performers.

The autoencoder of Brunner et al. (SQ11)[34] is slightly related to the research topic. The use of polyphonic is not relevant. The autoencoder is a good method to use when there is not enough labeled data available. This can be the case for the task of mid-level features extraction (**RQ1**). In that case, pre-trained models from other domains can also be adapted.

The multimodal architecture of Chen and Li (SQ13)[35] is not particularly related to the research topic. Some of the methods or ideas they used can be adapted to performance style. However, lyrics cannot be used in the domain of performance style for piano. The classification method for audio does not highlight the deviations of a performer. Thus, it will likely produce poor results if it is adapted to the task of performance style classification.

The method of Hajj et al. (SQ14)[36] is not directly related to the research topic. It is not able to distinguish piano performances of the same piece. However, some parts of the method could be altered to fit the tasks of this thesis. The feature generation with n-grams (**RQ1**) and feature reduction with CA (**RQ2**) showed promising results.

The set of audio features, proposed by Weiss and Müller (SQ15)[37], is related to the task of extracting mid-level features (**RQ1**). Even though the features were not explainable for humans, they captured local style patterns of music. The extraction method was tested for historical periods, which is strongly related to composer style. This focus on musical structure might be too high level to be used in the domain of performance style.

The study of Chou et al. (SS11)[38] is related to most of the subdomains of Music Information Retrieval (MIR). It is possible to adapt their techniques to this thesis' tasks. The comparison of tokenization methods (REMI, CP) could indicate what format is best for machines to understand the underlying structure of music. This is one of the challenges related to the extraction of mid-level features (**RQ1**).

The study of Chowdhury et al. (SS13)[2] is strongly related to the method of this thesis. The overarching challenge of explainable performance recognition is divided into two research questions (**RQ1**, **RQ2**). However, this study discussed the consequences of the two step approach. It showed that the best results were achieved when these steps were combined and the models were trained jointly.

The statistical study of Chowdhury and Widmer (SS14)[39] tested lower level features' ability predict high level features like emotion. This is strongly linked to the overall goal of this thesis, where mid-level features will be used to predict performance style (**RQ1**, **RQ2**). The authors showed that the mid-level features have a better accuracy and are more explainable. Since the mid-level features are used for emotion recognition, they might not be able to distinguish individual piano performers.

The method of explainable models by Chowdhury et al. (SS15)[40] is related to the goal of understanding differences in musical performances. The authors used several techniques, including importance calculation of mid-level

features, to explain differences on local and global levels. These explanations could be used to find biases in datasets, model deficiencies or improve human understanding. The audioLIME method for explainability might not work in this thesis' tasks because of the required input.

3.4.3 Small Presence of Performance Style in Query Results

Only one study in the query result was in the subdomain of performance style. As discussed in subsection 3.1.1, this did not have to be a problem as long as it was possible to use snowballing to get to the most relevant research. Several studies pointed towards the domain review of computational models for performance style (SS4), which included a lot of highly relevant material. It was interesting to look at why most primary studies were not directly related to the research questions. One of the reasons might have been a result of the query being too broad. The 23300 results from the search query could indicate such an explanation. In addition, the domain of performance style is less researched than Music Emotion Recognition (MER). More researched domains increase the number of potential citations. This can result in a skewed query result, where most of the studies are linked to a single subdomain.

3.4.4 Recommended Studies not in Query Results

The fact that the recommended studies did not appear in the query results was not a good sign for the search engine, the query, or the relevance of recommended studies. Ordering studies by relevance is complex because the top studies should be new and highly cited studies. This is typically a trade-off since the number of citations is correlated with time. Yet, Google Scholar was seen as a trusted search engine and has often been used to solve this problem.

As discussed in subsection 3.4.3, the subdomain of performance style was not well represented in the query results. Since most of the recommended studies were in this subdomain or linked to it, they were less likely to be represented. The recommended studies could also be of lower relevance in the specified subdomain. For instance, the graph method of Daejeong et al. [14] was a novel approach to note embedding. The method will have a lower priority because of its lack of research history. The alignment tool of Nakamura et al. [15] is a part of the subdomain of music alignment, which was narrow and partly related to performance style. Since Nakamura et al. focused on timing deviations instead of performance style, the study broaden the perspective on how to capture and process music. Yet, its distance to the topic of style lowered its relevance.

4 Method

This chapter presents the methods used for answering the research questions. The collection of piano performances and labeling of mid-level perceptual features are described in section 4.1. The collected data is processed to be used for the tasks of this thesis. The processing methods are presented in section 4.2. The different model architectures used for the tasks are described in section 4.3. To answer the first research question, the prediction task of MLPFs is used. The classification task of pianists answers the second research question. The MLPF prediction- and pianist classification tasks are presented in section 4.4 and section 4.5, respectively.

4.1 Data Collection

Collecting data consisted of gathering performances of the same piece and labeling segments of these performances. Subsection 4.1.1 describes the gathering of performances from the e-competition website. This dataset is denoted as the pianist dataset used for the pianist recognition task. The collected performances were segmented and converted into audio files, which were labeled for MLPFs. The process of creating the MLPF dataset is described in subsection 4.1.2.

4.1.1 Piano Performance Dataset

The dataset of pianist performances was retrieved from the International Piano e-Competition [41]. This was the same resource that Rafee et al. used for their KDE model [24]. The competitors in the e-competition played several well-known pieces on Yamaha CFX concert grand pianos. These pianos were equipped with competition Yamaha Disklavier software which made it possible to capture all aspects of the performance and replay it on corresponding pianos. There were separate Schubert Sonata rounds in previous years of the competition. This made it possible to find multiple performances of the same piece and study performance style. As in the paper of Rafee et al., this thesis used Schubert's Sonata in B-flat Major, D960. There were 11 performers available on the website (year of participation in parentheses): Jean-Selim Abdelmoula (2018), Vasyl Kotys (2018), Cheng Guang (2014), Peter Friis Johannson (2014), Piotr Rozanski (2009), Eric Zuber (2009), Gregory DeTurck (2006), Konstantin Krasnitsky (2006), Mikhail Mordvinov (2006), Edisher Savitski (2006), Benjamin Kim (2004). Rafee et al. only used nine of the same performers and excluded Eric Zuber and Benjamin Kim. When reproducing the model of Rafee et al., the same nine performers were used. All of the performances were in midi format and downloadable as single files. Some of the performances were additionally divided into four movements, where each movement could be downloaded separately. All movements of all eleven pianists were used in the dataset.

4.1.2 Mid-Level Perceptual Feature (MLPF) Dataset

As of May 2022, the MLPF Dataset was neither completed nor published [42]. The project was being conducted at the time of writing this thesis by the College of Music and the Graduate School of Data Science coalition at SNU. However, there were plans to publish a paper on the topic and publicly make the dataset available. A brief explanation of the dataset's creation process is provided.

The goal of the MLPF dataset was to provide a framework for quanitfying the style of pianists. Prof. Jong Hwa Park and graduate student Jisoo Park from Seoul National University created a list of musical dimensions. After assessing the musical dimensions with a focus group of graduate students from the College of Music at SNU, the list was refined. Dependent and conditional dimensions were added for musical research. The dimensions could be viewed as continuous, one-dimensional vector spaces. In most cases, the words used to describe these dimensions mark the extremes of the dimension. In some cases, the terms are relative and mark only the direction of the extreme for that dimension. All the dimensions were linked to a category to make them specific and unambiguous. Each row in Table 11 corresponds to one MLPF dimensions. The values for the columns *Extreme 1* and *Extreme 2* defines the dimension, for the specific category. Synonyms were added for some of the dimensions. A term in the *Synonym 1* column is synonymous with the term in the *Extreme 1* column. The same relationship applies to *Synonym 2* and *Extreme 2*.

Performances of Schubert's Sonata (B-flat Major, D960) were segmented into segments of 8 bars. This was viewed as the shortest interval where all dimensions could be determined. Simultaneously, the length entailed a possibility of change in dimension values during the segment.

A Likert scale from one to seven was used to label the segments for all the dimensions. The extremes in Table 11 define the scale. *Extreme 1* corresponds to one on the scale, *Extreme 2* corresponds to a seven on the scale and four is in the middle. Users were also able to use 0 if they were not sure. Three conditional dimensions were shown based on other dimension answers of the user. If the user chose a value on the side of Stable (1-3) in the first dimension (Stable/Unstable Beat), the second dimension (Mechanical/Natural Tempo) would be shown. However, Unstable Beat (5-7) would show the third dimension (Intensional/Unintentional). The fourth dimension (Regular/Chaotic Beat Change) was only shown if the user answered Unstable (5-7) in the first dimension and Intentional (1-3) in the third dimension.

The labeling process was conducted by Crowdworks, an online crowdsourcing company for AI. The labelers were experienced musicians, with more than 10 years experience. All of the labelers had at least a bachelor's degree in music. At the time of writing, data was only been collected for the second movement of Schubert's Sonata and a selection of performances: The score version of the piece and the six first pianists (Abdelmoula, Kotys, Guang, Johannson, Rozanski, Zuber).

Intraclass correlation (ICC) was used to test the reliability of the dataset. The labelers are one sample of domain experts. Since the results of the ICC

Table 11: Dimensions of MLPFs

Category	Extreme 1	Extreme 2	Synonym 1	Synonym 2
Time (Beat / Tempo)	Stable	Unstable		
	Mechanical	Natural		
	Intentional	Unintentional		
	Regular Change	Chaotic Change		
Articulation	Long	Short		
	Cushioned	Solid	Soft	Hard
Pedal	Saturated	Sparse	Wet	Dry
	Clean	Blurred		
	Subtle Change	Obvious Change		
Tone	Even	Colorful		
	Rich	Shallow		
	Bright	Dark		
Intensity (Tone/ Contrast)	Pure	Dramatic		Expressive
	Soft	Loud		
	Sophisticated	Raw	Mellow	Crude
	Balanced	Unbalanced		
	Large Dynamic Range	Small Dynamic Range		
Music Making (Musicality)	Fast Paced	Slow Paced		
	Flowing	Choppy		Articulated
	Swing	Steady	Flexible	Inflexible
	Flat	Spacious		
	Harmonious	Disproportioned		
Emotion (Characteristic)	Optimistic	Pessimistic	Pleasant	Sad
	High Energy	Low Energy		
	Dominant	Subdued	Forceful	
Mood	Imaginative	Honest		
	Ethereal	Mundane		
Interpretation	Convincing	Unsatisfactory		Doubtful

analysis should be relevant for other samples of domain experts as well, the two-way random method was used. Both single score and the average score was tested to see if one label can be used on their own. The absolute agreement was used in both cases. If the scores were reliable for a two-way random method with a single score and absolute agreement, ICC(2,1), the labels could be used independently. If this was not the case, an aggregate of the labels must be used for the data to be considered reliable. This was tested by the two-way random method with average measures and absolute agreement, ICC(2,k).

4.2 Processing

This section describes the adaption of the datasets to the given tasks. The performances of the pianist dataset are aligned and the deviations of each performer are computed. These processes are described in subsection 4.2.1. The labels from the MLPF dataset are aggregated for better reliability. The aggregation method is presented in subsection 4.2.2.

4.2.1 Audio Processing

To compare the performances of several pianists, the performances were aligned to the score. This was achieved by using the alignment tool of Nakamura et al. [15]. To align the score in the MIDI version with itself, the settings of the alignment tool needed to be configured. The threshold setting of the *ScorePerformMatcher* program was set to zero. The alignment of a performance to a score outputted a *match* file with low-level features: ID, onset time, offset time, spelled pitch, onset velocity, offset velocity, channel, match status, score time, note ID, error index and skip index.

The preprocessing of low-level features from the match files was based on the work of Rafee et al. [24]. Four low-level features were used directly: Onset time, offset time, onset velocity and offset velocity. The duration of a note was computed by subtracting the onset time from the offset time. IOI was calculated by subtracting the previous note's onset time from the current note's onset time. OTD was computed by subtracting the offset time of the current note from the onset time of the following note. The computed features are shown in Figure 11.

Unmatched and duplicated notes from the alignment were removed. The note IDs are used to group notes by different pianists and compute the average note. The method of using deviations is described by Stamatatos [3] and Rafee et al. [24]. The deviations of each pianist from both the score and the average performance were calculated. These deviations were used as inputs to the models.

4.2.2 MLPF Processing

The MLPF dataset contains a distribution of answers by different users. Three different methods were tested to create a ground truth label for all combinations of performers and segments. First, mean and median values for the distributions were calculated. Second, KDE was used, where the apex of the curve was considered as the ground truth. The KDE method was used with a bandwidth of 1 and a gaussian kernel. The distribution curve was computed using 100 steps. Discussions with prof. Jong Hwa Park and Jisoo Park revealed that the apex ground truth method was considered the best approximation of the ground truth label. The methods are visualized in Figure 12.

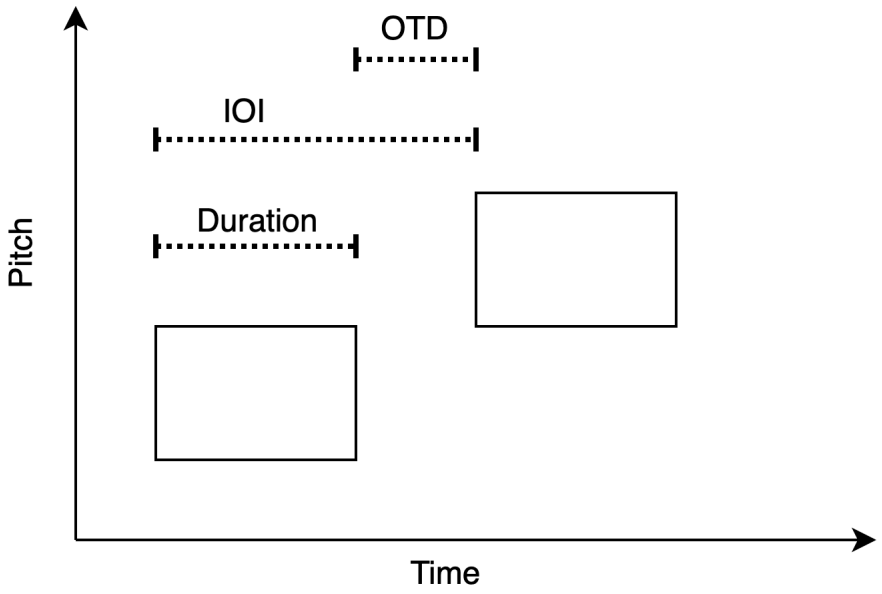


Figure 11: Computed Note Features

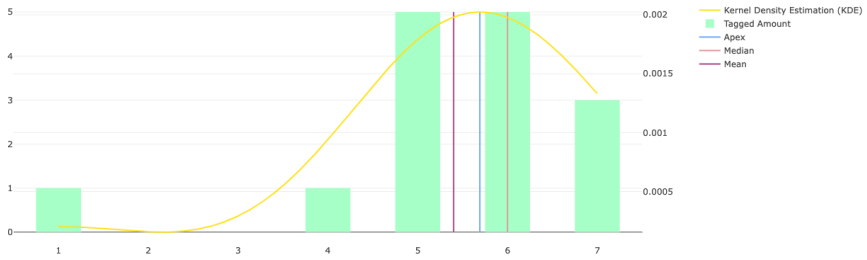


Figure 12: Ground Truths Calulation, Before Normalization

4.3 Models

The model architectures that were introduced in section 1.2 are described in detail in this section. Models that have the N2-abbreviation are novel. These models were based on recurrent neural networks and linear layers for predictions. However, the model based on Kernel Density Estimation was introduced by Rafee et al. [24].

4.3.1 Notes-to-MLPFs (N2M)

Figure 13 shows the architecture of the N2M model. The models included a dropout layer to improve their ability to generalize on unseen data. In addition, fully connected linear layers were added to transform the output of the RNN-cells to the correct output dimensions. RELU activation functions were used for the linear layers.

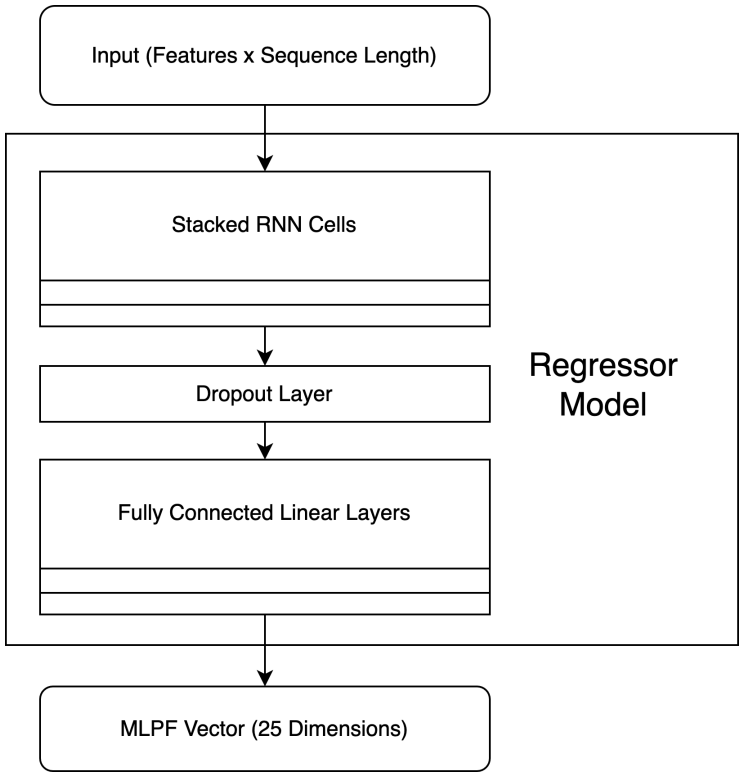


Figure 13: RNN Regressor Architecture

The training loop consisted of training and validation on the MLPF dataset. The Optuna software was used to tune hyperparameters and find the best-performing models. The search space was defined for all parameters. Log scale was used when the search space had significant differences in value sizes and

where minor differences near the lower bound should be studied as closely as significant differences near the upper bound. The random seed was added to minimize the probability of unfortunate initializations of weights or ordering of batches. Different combinations of deviation type and RNN type were trained in separate studies. Both the best intermediate model (*Int*) and the best model of the last epoch were saved (*End*) based on the score on the validation set. The following hyperparameters were tuned to reach the best results for the regressor:

- Deviation type: Score, Average
- RNN Type: GRU, LSTM
- RNN Hidden size: 10 - 1000, log-scale
- RNN Layers: 1-5
- Linear Layers: 1 - 5
- Linear Nodes: 10 - 1000, log-scale
- Learning Rate: 1e-6 - 1e-2, log-scale
- Random Seed: 0 - 10
- Batch Size: 10 - 60
- Dropout: 0 - 1

4.3.2 Notes-to-MLPFs-to-Pianists (N2M2P)

The architecture of the regressor model was identical to the regressor architecture of the N2M model. Figure 14 shows that a classifier model was added to the pipeline to classify pianists based on the MLPFs. This module consisted of fully connected linear layers and a softmax layer. The regressor models of the MLPF prediction task were reused. The weights of these models were frozen so that only the classifier model was trained.

To reduce computation, an intermediary dataset was created using the regressor model. The input values of the dataset were the MLPF output of the regressor model and the outputs were the correct pianists. Classifier models were trained using the different RNN models trained in the MLPF prediction task. This includes all combinations of the RNN types (LSTM, GRU), deviation types (Score, Average) and save times (Intermediary, End). Optuna optimized the following hyperparameters:

- Linear Layers: 1 - 5
- Linear Nodes: 10 - 1000, log-scale
- Learning Rate: 1e-6 - 0.1, log-scale
- Random Seed: 0 - 10

- Batch Size: 10 - 60

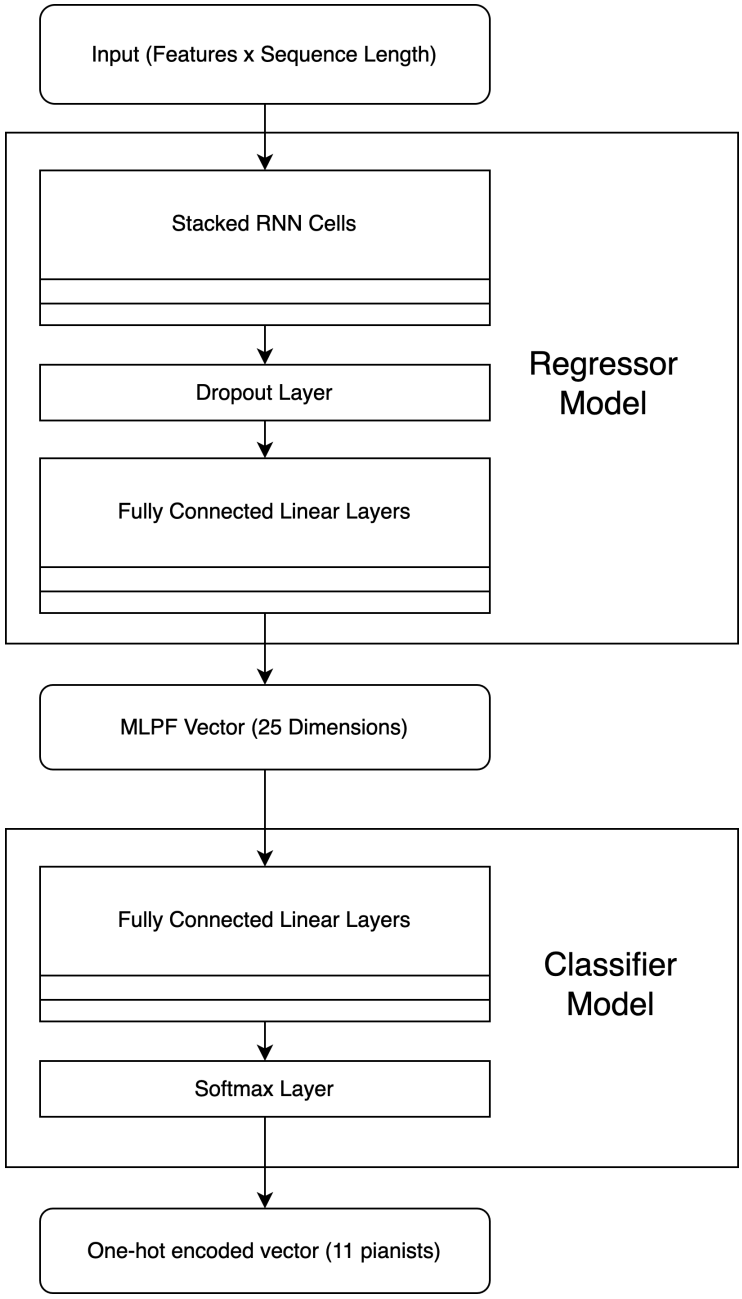


Figure 14: RNN Classifier via MLPFs (N2M2P, N2MP)

4.3.3 Notes-to-MLPFs-and-Pianists (N2MP)

The N2MP architecture was identical to the N2M2P architecture (see Figure 14). The only difference is that both the regressor and the classifier of the model were trained jointly. The N2MP approach used both the MLPF dataset and the pianist dataset in the same training loop. Each training loop started with the regressor model and the classifier model trained using the pianist training set. Then, the regressor model was trained using the MLPF training set. The specific order of training was set because of the large differences in dataset sizes. Since the MLPF dataset was smaller than the pianist dataset, the MLPF training was put last to ensure an impact. The pianist training set had 20872 samples, while the MLPF training set had 92 samples. Both the regressor and classifier models were finally validated using the validation sets. In the N2MP case, none of the weights were frozen during training.

Optuna optimized the following hyperparameters in the N2MP case:

- Deviation type: Score, Average
- RNN Type: GRU, LSTM
- RNN Hidden size: 10 - 1000, log-scale
- RNN Layers: 1 - 5
- Linear Layers: 1 - 5
- Linear Nodes: 10 - 1000, log-scale
- N2M Learning Rate: 1e-6 - 0.1, log-scale
- N2P Learning Rate: 1e-6 - 0.1, log-scale
- Random Seed: 0 - 10
- Batch Size: 10 - 60
- Dropout: 0 - 1

Since two metrics evaluated the N2MP approach simultaneously, it was difficult to evaluate the best performing models during training. Therefore, the scores were saved instead of the models. After training, the best performing models were retrained with the same parameters. Even though the parameters are the same, the results could be different after retraining. This is a result of the non deterministic parts RNNs in the pytorch library [8]. The best performing models were at the score frontier of the two dimensional metrics plot. The score frontier is defined by the coordinates that do not have better results in both dimensions. Three models for each configuration was selected for retraining. All of these models were above -1 in R^2 score. One model was selected for its R^2 score on the prediction task, one for its accuracy on the pianist classification task and one for both. The models saved models were tested on the test set of the MLPF dataset.

4.3.4 Notes-to-Pianists (N2P)

For the N2P approach, the architecture of the RNN models were similar to the N2M models. As Figure 15 shows, only a softmax layer was added to the model. In addition, the size of the output vector was changed to 11, to reflect the output of the recognition task.

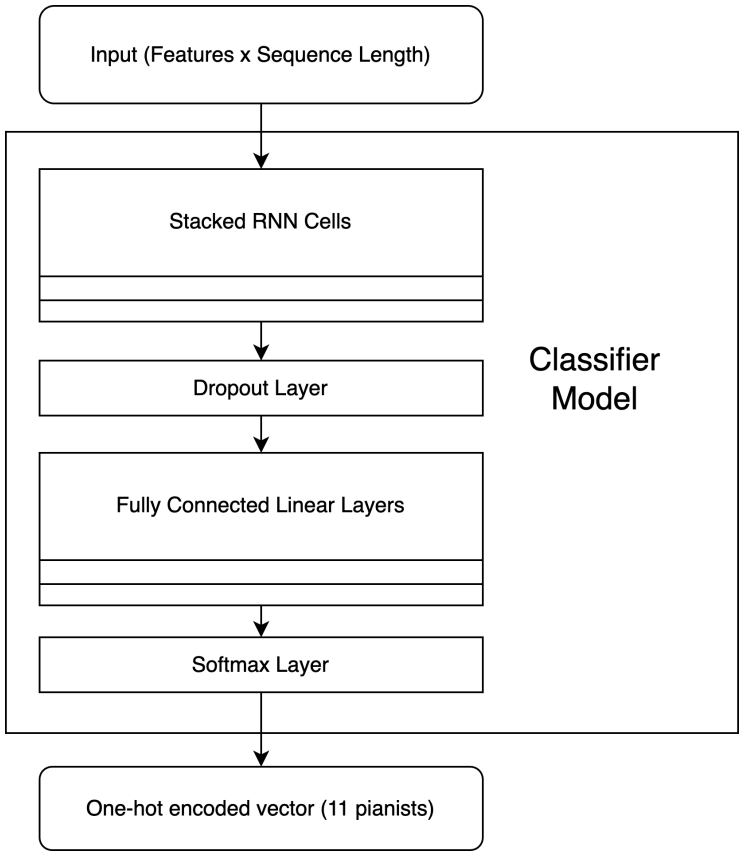


Figure 15: RNN Classifier Architecture (N2P)

4.3.5 Kernel Density Estimation (KDE) - Reproducing

The model based on Kernel Denisty Estimation by Rafee et al. [24] was used to compare the results of the RNN based models. The model captured the distributions of the pianists' low-level features. When classifying the pianist of a set of notes, the KL-divergence between the known performer distributions and the unknown pianist were calculated. The divergences of all features were summed, either weighted or with equal weights. The unknown performer was classified as the performer with the smallest summed divergence.

To test that the KDE model was correctly reproduced, it was tested in the same way as the original paper: An 8-fold cross validation scheme. Rafee et al. viewed the notes as independent data points. This approach resulted in folds that contained notes at random placements throughout the piece.

Precision, recall and f1 score were used to measure the models' performances. Sklearns package was used to compute these scores [43]. The original paper of Rafee et al. did not specify which method of averaging the classes was used for the score functions [24]. The argument was needed in the case of multilabel classifaction tasks. The scores in this thesis was calculated with the *'micro'* configuration for the average parameter. This computation method calculated the metrics globally. Confusion matrices were used to further explain the models' performances.

A model with the same architecture as the reproduced model was optimized using Optuna. The hyperparameters that were optimized are *n samples* and *bandwidth*. *Bandwidth* defines the size of the kernel. *N samples* determines the amount of values that the log-likelihood function will be computed for. *N samples* was added as a tunable hyperparameter. The paper of Rafee et al. [24] discussed the distributions, but did not state how many values they used for the distributions. Since the parameter had to be added to create the distributions, a wrongly chosen value could impact the results negatively. A high value for *n samples* correspond to a more fitted distribution. In the case of two configurations with the same score, the one with the smallest value for *n samples* was chosen. This distribution was considered to be less fitted and more appropriate for generalization.

Two sets of features were tested for the original and the optimized models. Using all the features did not produce the best results in the original paper of Rafee et al. [24]. The combination that performed the best was: IOI+DL+ND. *IOI* denotes Inter Onset Interval. DL is short for *Dynamics Level*, which corresponds to the Onset Velocity feature in this thesis. ND is short for *Note Duration*, which is denoted Duration in this thesis. The other model used all the features: OT+IOI+OTD+DL+ND. *OT* and *OTD* are short for Onset Time and Offset Time Duration.

4.4 MLPF Prediction Task

The MLPF prediction task was conducted by quantitatively testing different models on the ground truth data of the MPLF dataset. The task is connected to the first research question: what are the best methods for extracting MLPFs? The MLPF data is divided into three sets. The training set consisted of 80% of the data, 10% for validation and 10 % for testing. The evaluation metric for the task was the coefficient of determination (R^2 score). The Adam optimizer and Mean Square Error loss function were used for model training.

4.4.1 Input and Output

Since the MLPFs are based upon relationships between the notes, these relationships needed to be present in the input data. Therefore, this prediction task used sequences of notes as input. This is in contrast to Rafee et al., who used notes independently of their context [24]. To speed up the training process with data batching, all the sequences needed to be of equal lengths. The shortest sequence size in the MLPF dataset, 78, was used. Longer sequences are cut so that only the last 78 notes are used for the input data. As discussed in subsection 2.2.2, the recurrent nature of RNN models will place higher importance on the last input data. Therefore, the final notes of the segments were used. The input data consisted of either the deviations from the average performance or the deviations from the score.

The output was a vector with 25 values in the interval from -1 to 1. The three conditional dimensions of the MLPF dataset, described in subsection 4.1.2, were excluded from the multi-output prediction. These three dimensions comprised of 81.6%, 91.5% and 97.1% nan values, respectively. The output data was normalized from the Likert scale of 1-7 to a scale from -1 to 1. The normalized scale was used to convey the duality of the MLPF dimensions, where 0 was supposed to be the default value.

4.4.2 Models

The N2M and N2MP models was compared to the baseline models and humans. There were two baseline models to be tested. The first baseline model predicted the central value (0 when using the normalized MLPF dataset, called *Center*) for all inputs and dimensions. The second model predicted the average of each MLPF dimension based on the samples in the training dataset (*Average Training*). These models did not provide differences in style output and were only meant as benchmarks for model and human performances. Each labeling participant of the MLPF dataset contributed to the underlying data of the apex ground truths. Even though the participants contributed to creating the apex ground truths, they was also measured on how well their labels predicted the derived ground truths. The RNN based models were trained for 100 epochs.

4.5 Pianist Classification Task

The pianist classification task was conducted by evaluating the classification accuracy of different models on the pianist dataset. This task is connected to the second research question concerning pianist classification based on MLPFs. As in the prediction task, the model's input was sequences of subsequent notes. The output was, in contrast, one-hot encoded vectors representing the eleven pianists. The evaluated models include a baseline model, the reproduced KDE-, N2P-, N2M2P- and N2MP models.

4.5.1 Input and Output

Chunking was used to create sequences of notes for input to the models. The chunking consisted of a sliding window with a given offset and size that grouped together individual notes. In this task, the window size was set to 78, corresponding to the sequence size of the MLPF prediction task. The window offset determines how far the window will be moved each time. This parameter was used to increase the number of data sequences acquired from the same underlying sequence of data. If the window offset was set to the window size value, there would be no overlap in the data sequences. The window offset was set to 10 to maximize the amount of data extracted while not having too similar sequences. The data obtained from chunking could not be divided into training-, validation- and test sets. The overlap of sequences would lead to data leakage, where the same notes would potentially be present in several sets. Therefore, a performance was divided into non-overlapping segments before chunking was conducted. The same ratio of sets was used as in the MLPF prediction task: 80% for training, 10% for validation and 10% for test. For the KDE models, the training data was not chunked. This is because the model is based on creating distributions for each pianist and computing the entropy of each unknown sample. Therefore, the known pianist distributions were calculated based on all the notes in the training set. However, the validation and test sets were chunked to test the models fairly.

4.5.2 Models

The N2M2P-, N2MP- and N2P models was compared to a baseline and the reproduced KDE models. The baseline model predicted the pianist most represented in the training dataset. The first KDE model used the same parameter configuration as in the original paper of Rafee et al. [24]. The hyperparameters of the second KDE model were tuned using the performance of the validation set.

5 Results

This chapter presents the results of the models on each task and the results of the data collection and processing. The reliability and data sizes for the MLPF dataset are shown in section 5.4. Results of the alignment process are presented in section 5.2. Reproducing of the Kernel Density Estimation (KDE) model is presented in section 5.3. The results of each model on the MLPF prediction task and the pianist classification task are shown in section 5.4 and section 5.5, respectively.

5.1 Mid-Level Perceptual Features (MLPF) Dataset

The dataset consists of 1690 responses. Each response includes the labels for 28 dimensions. There are a total of 111 unique combinations of segments and pianists. When this is divided into training-, validation- and test sets of 80%, 10% and 10%, the resulting sample sizes are 92, 11 and 12, respectively. Table 12 shows the number of labeled segments for each user. The users are ordered by their amount of labels, in descending order. Two users were excluded from the table because their user ID included their email addresses. These two users had 62 and 1 label, respectively.

Table 12: Amount of Labeled Segments for Each User in the MLPF Dataset

User	Labels
1	115
2	115
3	115
4	114
109	70
97	70
70	70
101	70
113	70
28	70
71	70
86	70
114	70

User	Labels
90	59
0	53
121	51
120	45
116	39
84	28
96	27
100	22
102	22
104	21
110	20
117	20
78	19

User	Labels
73	15
93	15
95	14
77	13
103	12
87	11
85	10
6	10
108	7
107	4
91	1

Table 13 and Table 14 presents the two ICC methods for all the dimensions of the MLPF dataset. The differences in amounts of labels shown in Table 12 are incompatible with ICC computation for all users. The four users with the most labels produced a compatible set for the calculation of ICC. The columns *Lower Bound* and *Upper Bound* corresponds to the lower and upper bound of the 95% confidence intervals.

Table 13: ICC(2.1) for Each Dimension of the MLPF Dataset

Category	Extreme 1	Extreme 2	ICC(2.1)	Lower Bound	Upper Bound
Time (Beat / Tempo)	Stable	Unstable	0.09	0.02	0.17
Articulation	Long	Short	0.17	0.08	0.27
	Cushioned	Solid	0.10	0.03	0.19
Pedal	Saturated	Sparse	0.39	0.3	0.5
	Clean	Blurred	0.28	0.17	0.4
	Subtle Change	Obvious Change	0.09	0.02	0.19
Tone	Even	Colorful	0.14	0.06	0.23
	Rich	Shallow	0.15	0.07	0.25
	Bright	Dark	0.35	0.24	0.47
Intensity (Tone/ Contrast)	Pure	Dramatic	0.14	0.06	0.24
	Soft	Loud	0.25	0.16	0.35
	Sophisticated	Raw	0.25	0.13	0.37
	Balanced	Unbalanced	0.17	0.08	0.27
	Large Dynamic Range	Small Dynamic Range	0.18	0.09	0.29
Music Making (Musicality)	Fast Paced	Slow Paced	0.32	0.22	0.42
	Flowing	Choppy	0.13	0.05	0.22
	Swing	Steady	0.18	0.1	0.28
	Flat	Spacious	0.26	0.17	0.37
	Harmonious	Disproportioned	0.09	0.02	0.18
Emotion (Characteristic)	Optimistic	Pessimistic	0.35	0.22	0.48
	High Energy	Low Energy	0.24	0.15	0.35
	Dominant	Subdued	0.16	0.08	0.26
Mood	Imaginative	Honest	0.18	0.09	0.29
	Ethereal	Mundane	0.15	0.07	0.25
Interpretation	Convincing	Unsatisfactory	0.14	0.05	0.24

Table 14: ICC(2,k) for Each Dimension of the MLPF Dataset

Category	Extreme 1	Extreme 2	ICC(2,k)	Lower Bound	Upper Bound
Time (Beat / Tempo)	Stable	Unstable	0.28	0.07	0.45
Articulation	Long	Short	0.45	0.27	0.6
	Cushioned	Solid	0.31	0.11	0.48
Pedal	Saturated	Sparse	0.72	0.63	0.8
	Clean	Blurred	0.61	0.45	0.73
	Subtle Change	Obvious Change	0.29	0.07	0.48
Tone	Even	Colorful	0.39	0.19	0.55
	Rich	Shallow	0.42	0.22	0.58
	Bright	Dark	0.69	0.56	0.78
Intensity (Tone/ Contrast)	Pure	Dramatic	0.40	0.2	0.56
	Soft	Loud	0.57	0.43	0.69
	Sophisticated	Raw	0.57	0.38	0.7
	Balanced	Unbalanced	0.45	0.27	0.59
	Large Dynamic Range	Small Dynamic Range	0.48	0.27	0.63
Music Making (Musicality)	Fast Paced	Slow Paced	0.65	0.53	0.74
	Flowing	Choppy	0.37	0.17	0.53
	Swing	Steady	0.47	0.3	0.61
	Flat	Spacious	0.59	0.45	0.7
	Harmonious	Disproportioned	0.29	0.08	0.46
Emotion (Characteristic)	Optimistic	Pessimistic	0.69	0.53	0.79
	High Energy	Low Energy	0.56	0.41	0.68
	Dominant	Subdued	0.44	0.26	0.59
Mood	Imaginative	Honest	0.47	0.28	0.62
	Ethereal	Mundane	0.42	0.23	0.58
Interpretation	Convincing	Unsatisfactory	0.39	0.18	0.55

5.2 Processing

This section presents the statistics for the processing of audio data. Subsection 5.2.1 shows the frequencies of alignment statuses for all performers in the pianist dataset. Matched notes were used to calculate the average notes. The amount of notes that were used for this computation is shown in subsection 5.2.2.

5.2.1 Alignment

Figure 16 shows an excerpt of the alignment result of Abdelmoula and the score. The result was visualized using the alignment website of Nakamura et al. [15]. The yellow color indicates error regions. Blue notes correspond to added notes by the performer. Pink notes correspond to missing notes. Missing notes are in the score but are not played by the performer.

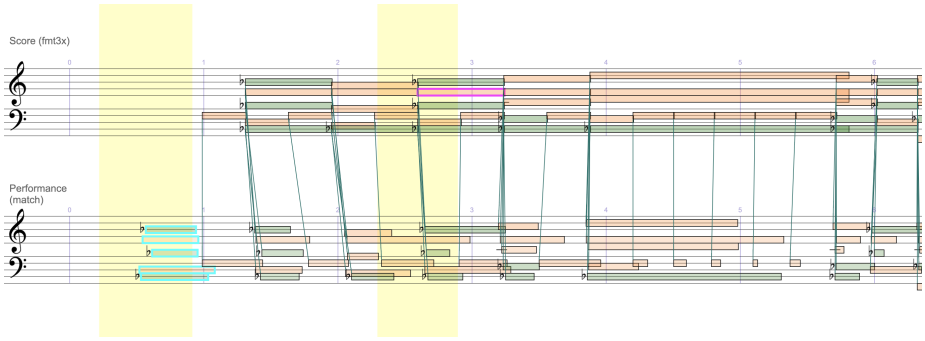


Figure 16: Alignment of Abdelmoula and Score

The results of the alignment process are shown in Table 15.

Table 15: Aligned Notes for Pianist Dataset

Performer	Matched	Added	Erroneous	Missing
Score	20644	1504	109	314
Johannson	18931	2249	341	1365
Kotys	18281	2659	495	1965
Abdelmoula	18240	2697	483	1952
Guang	15989	2439	448	4342
Kim	15834	2530	429	4371
Zuber	15826	2576	479	4484
DeTurck	15763	2571	398	4514
Rozanski	15742	2658	374	4467
Krasnitsky	15736	2997	392	4545
Savitski	15733	2515	420	4534
Mordvinov	15646	2753	470	4569

5.2.2 Calculating the Average Performance

The process of calculating the average performance was conducted for all notes of the MLPF and pianist datasets. The results are shown in Table 16 and Table 17, respectively.

Table 16: Number of Notes per Average Note for the MLPF Dataset

Notes Per Average	Frequency	Share
6	2109	93.4%
5	91	4.0%
4	16	0.7%
3	8	0.4%
2	7	0.3%
1	26	1.2%

Table 17: Number of Notes per Average Note for the Pianist Dataset

Notes Per Average	Frequency	Share
11	13794	71.1%
10	1452	7.5%
9	366	1.9%
8	171	0.9%
7	111	0.6%
6	56	0.3%
5	39	0.2%
4	38	0.2%
3	2544	13.1%
2	209	1.1%
1	623	3.2%

5.3 Reproduced Kernel Density Estimation (KDE)

This section presents the results of the reproduced Kernel Density Estimation models on the original classification task of Rafee et al. [24], using k-fold testing. The results for both the original parameters and optimized parameters are provided in subsection 5.3.1 and subsection 5.3.2 respectively. Confusion matrices for the models are presented in the appendix (A2.1).

5.3.1 Original Parameters

Table 18 shows the scores of the different feature combinations with original parameters.

Table 18: Scores for KDE models with Original Parameters

Model	Precision	Recall	F-Score
All features	0.444	0.444	0.444
Selected	0.458	0.458	0.458

Figure 17 presents the model of KDE with only the selected features. The figure shows that all the pianists have been classified as Abdelmoula but to varying degrees. Each of the eight folds included one sample for each of the nine pianists. Abdelmoula was predicted to be the unknown pianist 35 times. The total number of predictions was 72.

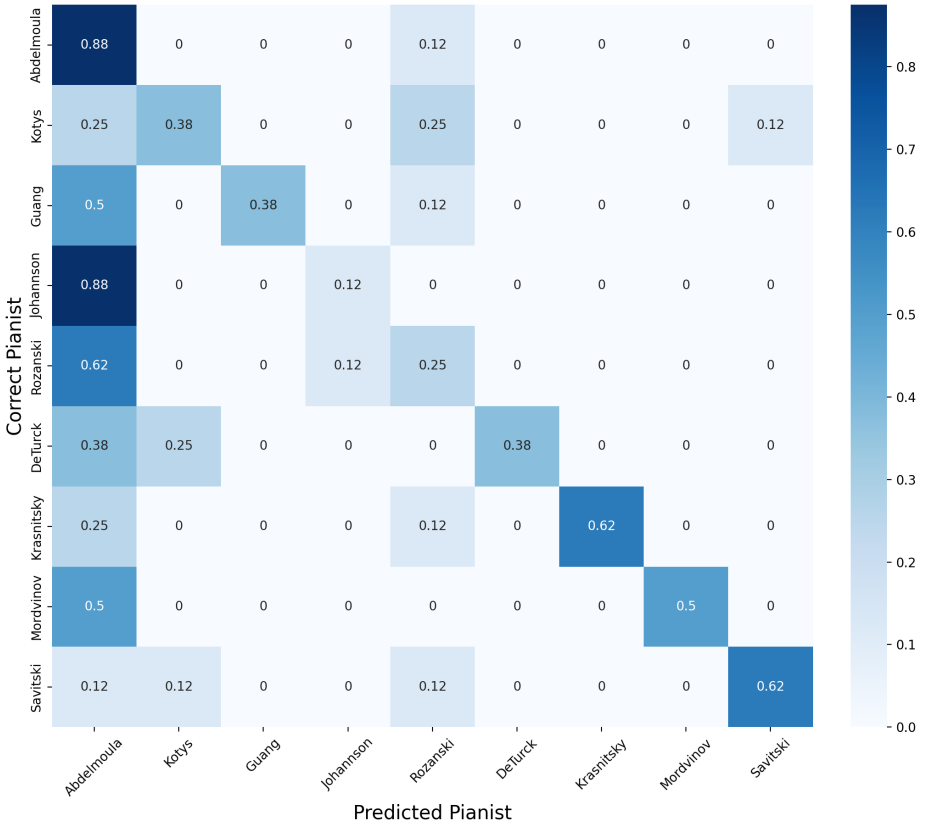


Figure 17: Original KDE with Selected Features

Figure 18 shows the onset time distributions of the correct classification of Abdelmoula. The test distribution of Abdelmoula is computed from the first fold and is shown in light green. Abdelmoula's training distribution is computed from the last seven folds and is shown in dark green. The right-hand side of the figure shows the known pianist distributions. Next to the pianists' names are the computed entropies between the known pianist distributions and the unknown pianist sample. In this case, all of the computed entropies were infinite.

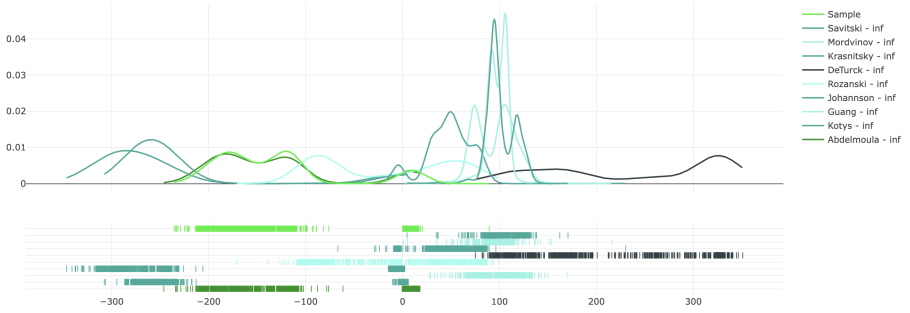


Figure 18: Onset Time Distributions, the Unknown Pianist Sample is from the 1st fold of Abdelmoula

5.3.2 Optimized Parameters

The optimized parameters for each of the individual feature dimensions are shown in Table 19. The KDE method can perfectly predict all the pianists by only using the onset time feature.

Table 19: Optimized Parameters for Single Feature Dimensions

Feature	Bandwidth	N samples	Precision	Recall	F1
Time Onset	33.96	5	1.00	1.00	1.00
Velocity Onset	1.53	76	0.96	0.96	0.96
Duration	0.29	29	0.86	0.86	0.86
Inter Onset Interval	0.05	315	0.50	0.50	0.50
Offset Time Duration	0.17	61	0.71	0.71	0.71

The distributions and entropies for each feature dimension are shown in Figure 19, Figure 20, Figure 21, Figure 22 and Figure 23. These are examples from the first fold, where Abdelmoula is the unknown pianist. Abdelmoula is correctly classified for all the feature dimensions except offset time duration. In the case of offset time duration, the divergence is the smallest between the sample distribution and Kranitskiy's distribution.

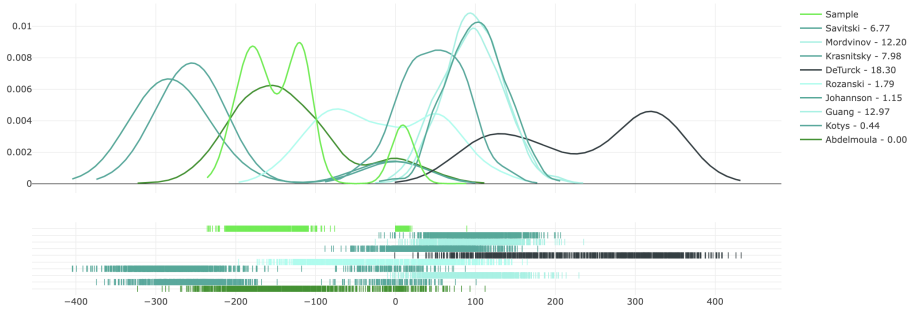


Figure 19: Pianist Distributions for Onset Time

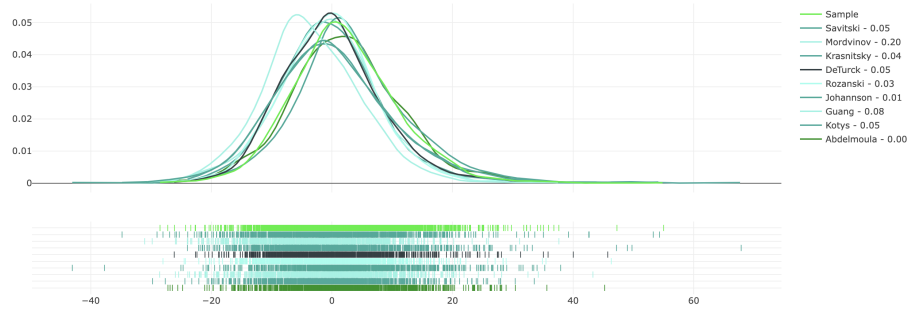


Figure 20: Pianist Distributions for Onset Velocity

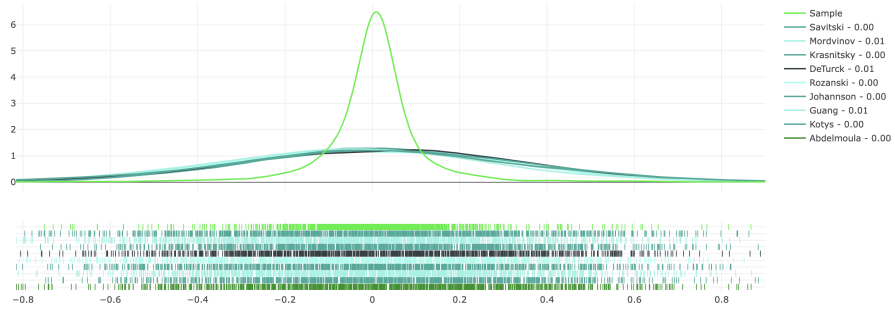


Figure 21: Pianist Distributions for Duration

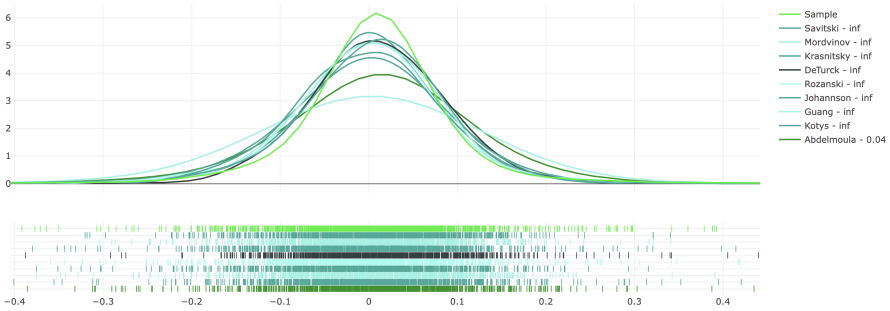


Figure 22: Pianist Distributions for Inter Onset Interval

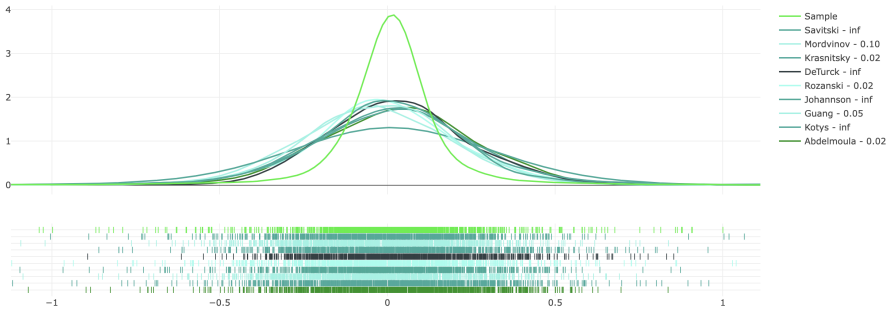


Figure 23: Pianist Distributions for Offset Time Duration

Table 20 shows the scores for the KDE model with optimized parameters on the k-fold classification task.

Table 20: Scores for KDE models with Optimized Parameters

Model	Precision	Recall	F1
All Features	0.67	0.67	0.67
Selected	0.42	0.42	0.42

5.4 MLPF Prediction Task

R2 scores measure the performance on the MLPF prediction task. These are presented in tables, one for each method. First, human and baseline scores are shown for all-, training-, validation- and testing data. Next, the method scores are shown for training-, validation- and test data. Finally, the relationships between ground truth labels and the predicted labels are added for the best performing models of each category. These relationships are visualized through scatterplots and represented by Pearson correlation coefficients. The model parameters for the best performing models are presented in the appendix (A3).

5.4.1 Baseline

The R^2-scores for the baseline models are displayed in Table 21. *Training Average* corresponds to the method of predicting the average of the training samples for all dimensions. The method of always choosing the center value (0 for the normalized MLPF dataset) is denoted as *Center*. The methods are ordered by their test score (*Test* column in Table 21) in descending order.

Table 21: Baseline Scores

Method	All	Training	Validation	Test
Average Training	0.00	0.00	–0.09	–0.13
Center	–0.17	–0.15	–0.40	–0.67

5.4.2 Humans

The R^2-scores for humans participating in the labeling of piano segments are shown in Table 22. In descending order, the participants are sorted by their score on the whole dataset (All).

Table 22: Human R^2 Scores

Participant	All	Training	Validation	Test
109	0.12	0.06	–0.98	0.18
86	0.03	0.04	–3.54	–0.39
116	–0.11	–0.15	–0.66	–1.86
114	–0.17	–0.22	–5.05	–1.79
121	–0.18	–0.11	–5.35	–14.07
113	–0.21	–0.10	–0.62	–1.71
93	–0.21	–0.62	–41.00	–9.43
120	–0.25	–0.20	–0.85	–8.08
3	–0.32	–0.30	–0.76	–0.47
90	–0.44	–0.39	–3.21	–1.81
71	–0.49	–0.52	–1.24	–4.69
28	–0.51	–0.34	–4.16	–47.96

78	–0.68	–0.74	–11.25	–8.20
97	–0.71	–0.61	–8.59	–1.11
0	–0.81	–0.99	–0.44	–0.98
4	–0.90	–0.83	–3.14	–1.46
70	–1.08	–1.00	–6.03	–1.25
1	–1.55	–1.48	–2.05	–2.89
2	–1.60	–1.53	–2.50	–2.97
73	–1.98	–2.33	–374.85	–6.67

Table 23 shows the different boundary values for confidence intervals at different confidence levels. The confidence intervals are computed using Student's T-distribution because of the small sample size.

Table 23: Confidence Intervals for Averaged Human R^2 Scores on All Data

Confidence Level	Average	STD	Upper	Lower
90%	–0.603	0.575	–0.381	–0.825
95%	–0.603	0.575	–0.334	–0.872
99%	–0.603	0.575	–0.235	–0.971

5.4.3 Notes-to-MLPFs (N2M)

The R^2-scores for models with the N2M-architecture are shown in Table 24. The *Save Time* column denoted when the model was saved. *Int* is short for Intermediate and corresponds to the model which performed the best on the validation data during training. *End* denotes the model saved after the last epoch of the training loop. The methods are ordered by test score, in descending order

Table 24: N2M Scores

RNN	Deviation	Save Time	Training	Validation	Test
GRU	Average	End	0.372	0.163	–0.007
LSTM	Average	Int	0.285	0.262	–0.009
LSTM	Average	End	0.461	0.209	–0.073
GRU	Average	Int	0.418	0.251	–0.202
GRU	Score	End	0.259	0.048	–0.307
LSTM	Score	Int	0.257	0.073	–0.350
GRU	Score	Int	0.261	0.104	–0.399
LSTM	Score	End	0.403	–0.015	–0.401

Table 25, Table 26 and Table 27 shows the average scores for each of the configurations for different confidence intervals. The confidence intervals are computed using Student's T-distribution because of the small sample size.

Table 25: Average Scores for N2M Models with 90% Confidence Interval

Configuration	Sample size	Average	STD	Lower	Upper
Average	4	–0.073	0.092	–0.180	0.035
LSTM	4	–0.208	0.196	–0.439	0.023
GRU	4	–0.229	0.168	–0.427	–0.031
Score	4	–0.364	0.045	–0.417	–0.311

Table 26: Average Scores for N2M Models with 95% Confidence Interval

Configuration	Sample size	Average	STD	Lower	Upper
Average	4	–0.073	0.092	–0.218	0.073
LSTM	4	–0.208	0.196	–0.520	0.104
GRU	4	–0.229	0.168	–0.496	0.039
Score	4	–0.364	0.045	–0.436	–0.293

Table 27: Average Scores for N2M Models with 99% Confidence Interval

Configuration	Sample size	Average	STD	Lower	Upper
Average	4	–0.073	0.092	–0.340	0.195
LSTM	4	–0.208	0.196	–0.781	0.365
GRU	4	–0.229	0.168	–0.720	0.262
Score	4	–0.364	0.045	–0.496	–0.233

5.4.4 Notes-to-MLPFs-and-Pianists (N2MP)

Table 28 shows the number of trials that passed the threshold of having an R2 score greater than -1. The results are listed by the different configurations of the models: all combinations of RNN type (GRU, LSTM) and deviation type (deviations from the average performance, deviations from the score).

Table 28: Training Results for N2MP

Model	Deviation	$R^2 > -1$	Completed Trials	Percentage
GRU	Average	137	353	38.8%
GRU	Score	142	306	46.4%
LSTM	Average	74	297	24.9%
LSTM	Score	175	313	55.9%

Figure 24, Figure 25, Figure 26 and Figure 27 show the frontiers of the N2MP models for each of the configurations. The selected models are indicated by yellow, orange and red data points. Each color corresponds to different reasons for selecting the model configuration. First, yellow entails that the model is chosen for its R2 score on the MLPF prediction task. Next, orange

indicates that the model is chosen because of its accuracy score on the pianist classification task. Finally, red entails that the model is chosen because of relatively good scores for both the MLPF prediction and pianist classification tasks. The scores of the models on the validation sets after training are shown next to the data points.

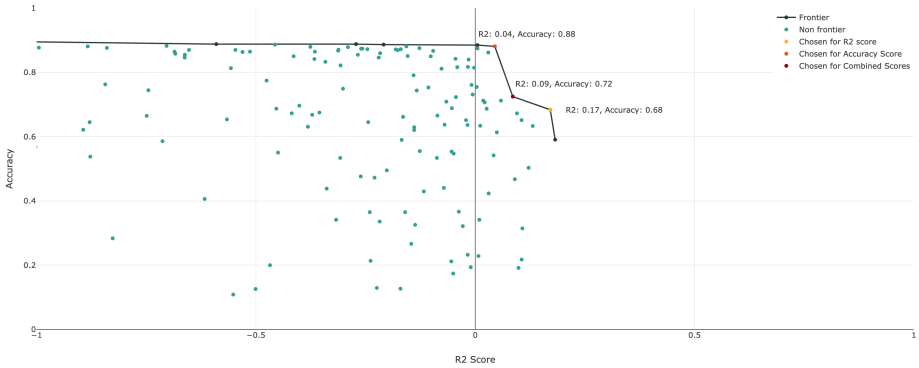


Figure 24: Training Results for GRU with Deviations from the Average

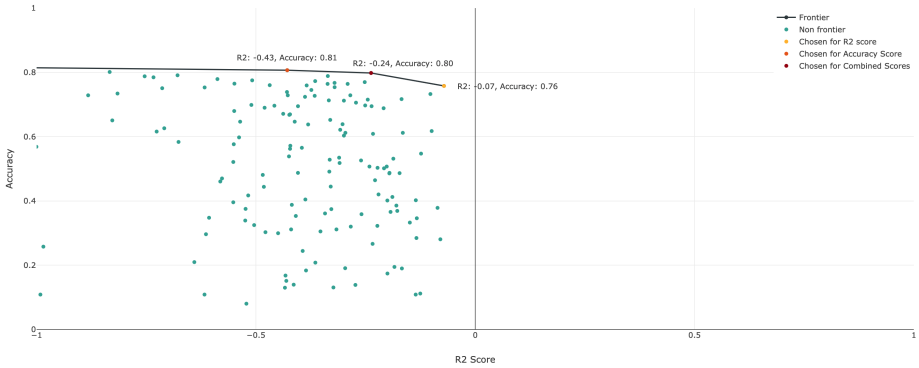


Figure 25: Training Results for GRU with Deviations from the Score

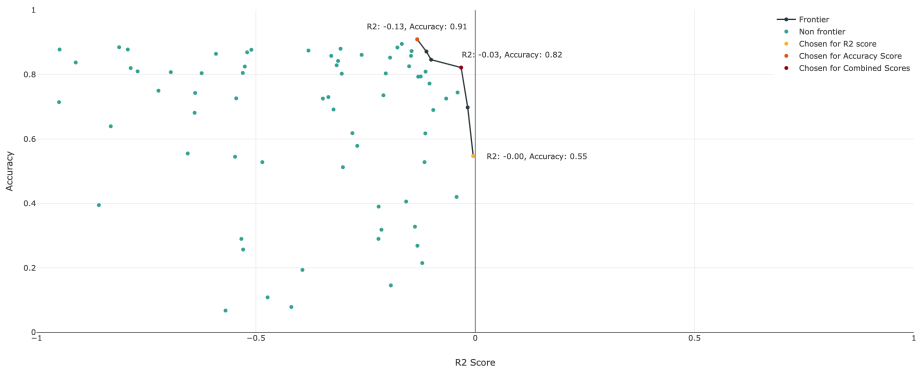


Figure 26: Training Results for LSTM with Deviations from the Average

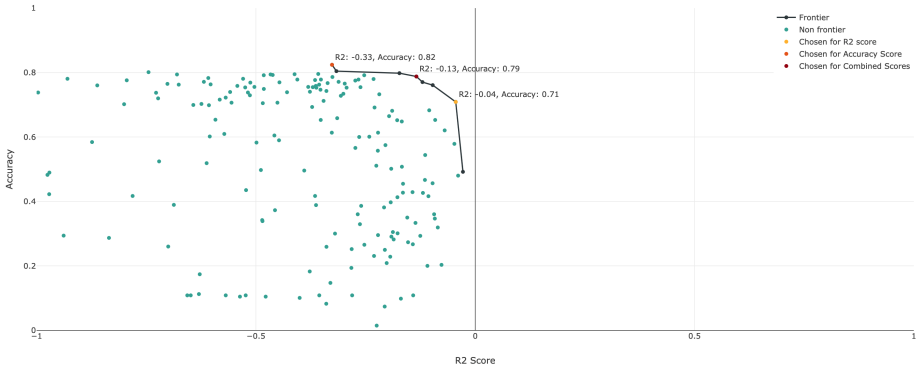


Figure 27: Training Results for LSTM with Deviations from the Score

The R^2-scores of the retrained models with the N2MP-architecture are shown in Table 29. The *Selection* column displays the reason for selecting the model for retraining. R2 corresponds to a model selected from the frontier based on its R2 score on the validation set of the MLPF dataset. *Accuracy* corresponds to a model that was selected based on its accuracy score on the validation set of the pianist dataset. *Combination* denotes a model selected because of good scores for both tasks. The *Save* column displays the reason for saving the model. R^2 corresponds to the model that performed the best on validation data of the prediction task for that given RNN type, deviation and trial. *Accuracy* corresponds to the model that performed the best on the validation data of the classification task for that given RNN type, deviation and trial. *End* denotes the model saved after the last epoch of the training loop. The methods are ordered by test score, in descending order.

Table 29: N2MP Scores

RNN	Deviation	Selection	Save	Training	Validation	Test
LSTM	Average	Accuracy	R^2	0.385	0.131	–0.014
GRU	Average	Accuracy	End	0.355	0.172	–0.042
LSTM	Average	Accuracy	Accuracy	0.313	–0.112	–0.052
GRU	Average	Accuracy	R^2	0.325	0.247	–0.054
GRU	Average	Combination	End	0.322	0.086	–0.064
GRU	Average	Combination	Accuracy	0.322	0.086	–0.064
GRU	Average	Combination	R^2	0.316	0.091	–0.070
GRU	Average	Accuracy	Accuracy	0.324	0.084	–0.081
LSTM	Average	Accuracy	End	0.394	–0.076	–0.166
LSTM	Average	Combination	End	0.552	–0.025	–0.293
LSTM	Average	Combination	Accuracy	0.541	0.062	–0.309
LSTM	Average	Combination	R^2	0.465	0.176	–0.323
GRU	Score	Accuracy	R^2	0.104	–0.142	–0.324
GRU	Score	R^2	R^2	–0.013	–0.165	–0.476
GRU	Score	Accuracy	Accuracy	0.658	–0.352	–0.592
LSTM	Score	Combination	R^2	0.265	–0.275	–0.641
GRU	Score	R^2	Accuracy	0.191	–0.454	–0.714
LSTM	Score	Combination	End	0.341	–0.430	–0.820
LSTM	Score	Combination	Accuracy	0.341	–0.430	–0.820
GRU	Score	Accuracy	End	0.668	–0.294	–0.833
GRU	Score	R^2	End	0.274	–0.321	–0.855
LSTM	Average	R^2	R^2	–0.034	–0.179	–1.288
LSTM	Average	R^2	End	0.039	–0.956	–1.488
GRU	Score	Combination	End	0.471	–0.148	–1.509
LSTM	Score	Accuracy	Accuracy	0.279	–0.241	–1.534
GRU	Score	Combination	R^2	0.477	–0.036	–1.641
GRU	Score	Combination	Accuracy	0.479	–0.082	–1.652
LSTM	Average	R^2	Accuracy	–0.119	–0.434	–1.707
LSTM	Score	Accuracy	End	0.340	–0.055	–1.903
LSTM	Score	Accuracy	R^2	0.340	–0.055	–1.903
LSTM	Score	R^2	End	0.268	–0.323	–2.320
LSTM	Score	R^2	Accuracy	0.251	–0.315	–2.596
LSTM	Score	R^2	R^2	0.141	–0.123	–2.616
GRU	Average	R^2	End	0.505	–0.211	–4.550
GRU	Average	R^2	R^2	0.402	0.058	–5.341
GRU	Average	R^2	Accuracy	0.459	–0.057	–5.730

The best models presented in Table 29 are used to calculate confidence intervals. The threshold set for a well-performing model is an R2 score larger than -1. All combinations of RNNs and deviations are represented with six models, except the combination of *LSTM* and *Score*. For the combination of *LSTM* and *Score*, three models perform better than the threshold. Table 27, Table 28 and

Table 29 present the average R2 scores and confidence intervals, with confidence levels of 90%, 95% and 99%, respectively.

Table 30: Average R^2 Scores for Best N2MP Models with 90% Confidence Interval

RNN	Deviation	Sample size	Average	STD	Lower	Upper
GRU	Average	6	–0.062	0.013	–0.073	–0.052
LSTM	Average	6	–0.193	0.137	–0.305	–0.080
GRU	Score	6	–0.632	0.208	–0.804	–0.461
LSTM	Score	3	–0.760	0.103	–0.934	–0.587

Table 31: Average R^2 Scores for Best N2MP Models with 95% Confidence Interval

RNN	Deviation	Sample size	Average	STD	Lower	Upper
GRU	Average	6	–0.062	0.013	–0.076	–0.049
LSTM	Average	6	–0.193	0.137	–0.336	–0.049
GRU	Score	6	–0.632	0.208	–0.851	–0.414
LSTM	Score	3	–0.760	0.103	–1.016	–0.505

Table 32: Average R^2 Scores for Best N2MP Models with 99% Confidence Interval

RNN	Deviation	Sample size	Average	STD	Lower	Upper
GRU	Average	6	–0.062	0.013	–0.084	–0.041
LSTM	Average	6	–0.193	0.137	–0.418	0.032
GRU	Score	6	–0.632	0.208	–0.976	–0.289
LSTM	Score	3	–0.760	0.103	–1.350	–0.171

5.5 Pianist Classification Task

This section presents the performance of each model to classify pianists, which is measured by accuracy. The accuracy scores are presented in tables, one for each method. In addition, confidence intervals are computed for the RNN models. Confusion matrices for the models are presented in the appendix (A2.2). The model parameters for the best performing models are presented in the appendix (A3).

5.5.1 Baseline

Johannson was the pianist with the highest representation in the training dataset. His shares of the training-, validation- and test sets are presented in Table 33.

Table 33: Baseline Scores

Method	Training	Validation	Test
Highest Representation	10.3%	10.9%	10.9%

5.5.2 KDE

The accuracy scores of the KDE models are presented in Table 34. Figure 28 and Figure 29 show the confusion matrices on the test data for the KDE models with all features for original and optimized parameters, respectively. These figures show that both versions can distinguish most of the performers. The samples of the pianists Abdelmoula, Kotys, Johannson, Rozanski, and DeTurck are perfectly classified for both models. There are significant decreases in scores when only using the selected features described in subsection 4.3.5.

Table 34: Accuracy Scores for KDE methods

Model	Features	Validation	Test
Original	All	72.9%	73.0%
Original	Selected	24.0%	27.0%
Optimized	All	76.2%	74.5%
Optimized	Selected	14.5%	12.4%

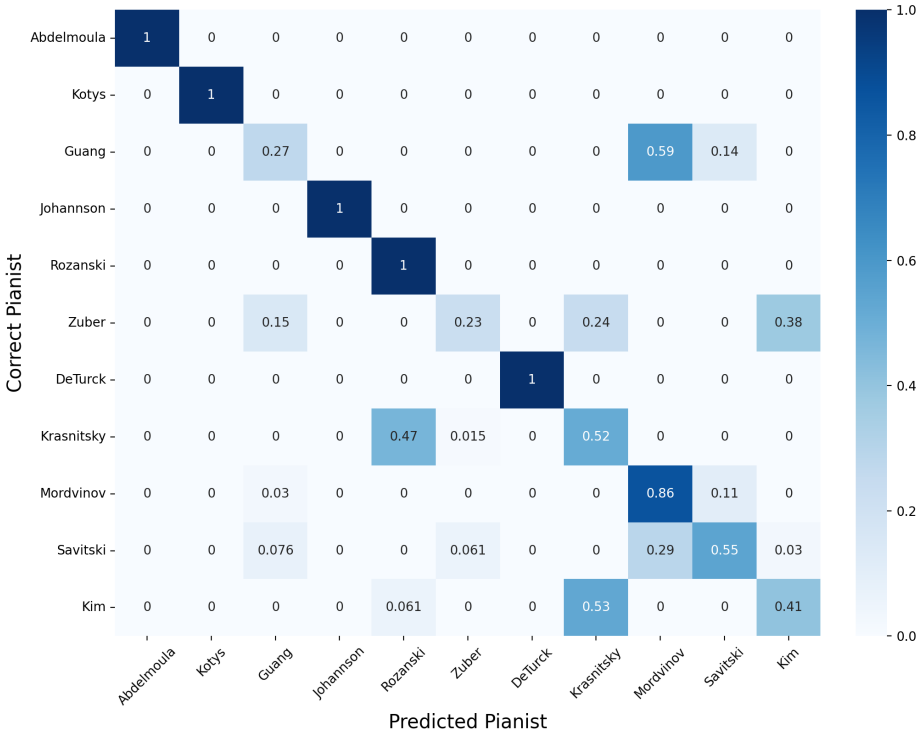


Figure 28: Original KDE with All Features

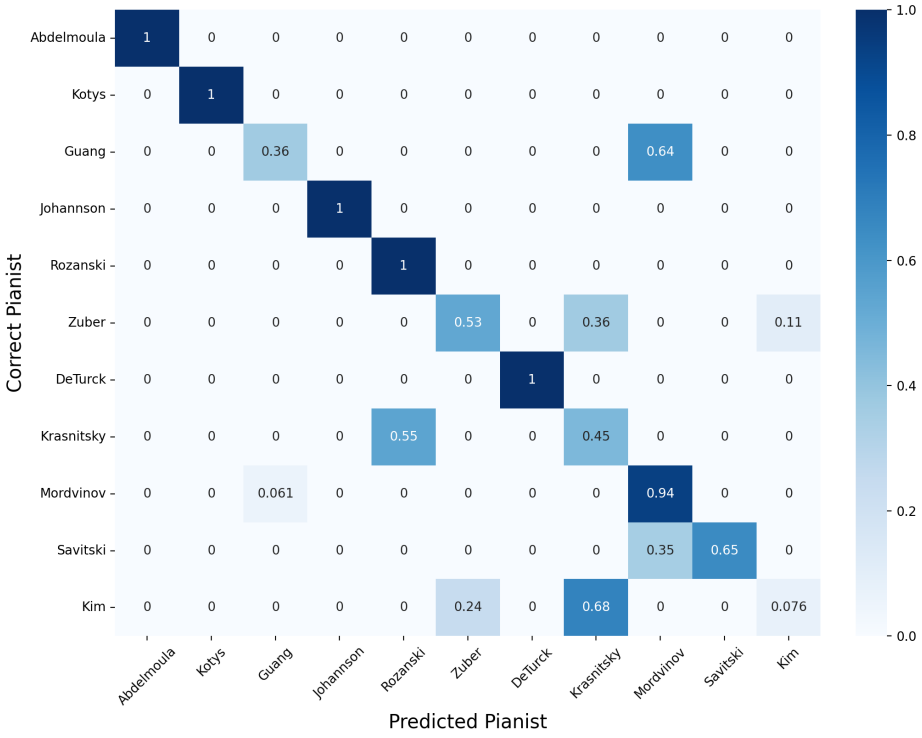


Figure 29: Optimized KDE with All Features

5.5.3 Notes to Pianists (N2P)

The accuracy scores of the N2P methods are presented in Table 35. The methods are ordered by their score on the test data (Test), in descending order.

Table 35: N2P Results

RNN	Deviation	Time	Train	Validation	Test
GRU	Average	End	99.3%	92.3%	82.8%
GRU	Average	Intermediate	99.5%	92.7%	81.3%
LSTM	Average	Intermediate	96.5%	91.7%	79.5%
LSTM	Average	End	99.5%	90.9%	77.4%
GRU	Score	Intermediate	91.6%	81.3%	71.2%
GRU	Score	End	99.4%	79.4%	69.8%
LSTM	Score	End	96.7%	81.8%	68.1%
LSTM	Score	Intermediate	96.7%	81.8%	68.1%

Table 36: Confidence Intervals for the 4 Best N2P Models (Deviation: Average)

Confidence Level	Average	STD	Lower	Upper
90%	0.802	0.023	0.775	0.830
95%	0.802	0.023	0.765	0.840
99%	0.802	0.023	0.734	0.871

Table 37: Confidence Intervals for the 4 Worst N2P Models (Deviation: Score)

Confidence Level	Average	STD	Lower	Upper
90%	0.693	0.015	0.675	0.710
95%	0.693	0.015	0.669	0.717
99%	0.693	0.015	0.649	0.737

5.5.4 Notes-to-MLPFs-to-Pianists (N2M2P)

The accuracy scores of the N2M2P models are presented in Table 38. The models are ordered by the average placement (*Avg No.*) in the prediction task (R^2 *No.*) and in the classification task (*Acc No.*).

Table 38: N2M2P Results

RNN	Deviation	N2M	M2P	R^2	Acc	R^2 No.	Acc No.	Avg No.
GRU	Avg	End	Int	–0.007	63.7%	1	1	1
GRU	Avg	End	End	–0.007	63.3%	1	2	1.5
LSTM	Avg	End	End	–0.073	42.2%	3	3	3
LSTM	Avg	End	Int	–0.073	40.6%	3	4	3.5
GRU	Avg	Int	Int	–0.202	38.9%	4	5	4.5
GRU	Avg	Int	End	–0.202	37.1%	4	6	5
LSTM	Avg	Int	End	–0.009	31.4%	2	8	5
GRU	Score	Int	Int	–0.399	31.8%	7	7	7
LSTM	Avg	Int	Int	–0.009	29.8%	2	12	7
GRU	Score	End	Int	–0.307	30.6%	5	10	7.5
GRU	Score	Int	End	–0.399	31.1%	7	9	8
GRU	Score	End	End	–0.307	29.3%	5	13	9
LSTM	Score	End	Int	–0.401	30.1%	8	11	9.5
LSTM	Score	Int	Int	–0.350	20.2%	6	15	10.5
LSTM	Score	End	End	–0.401	25.6%	8	14	11
LSTM	Score	Int	End	–0.350	20.0%	6	16	11

Figure 30 shows the correlation between the R^2 scores on the prediction task and the accuracy score on the classification task. The correlation between the scores is 0.67.

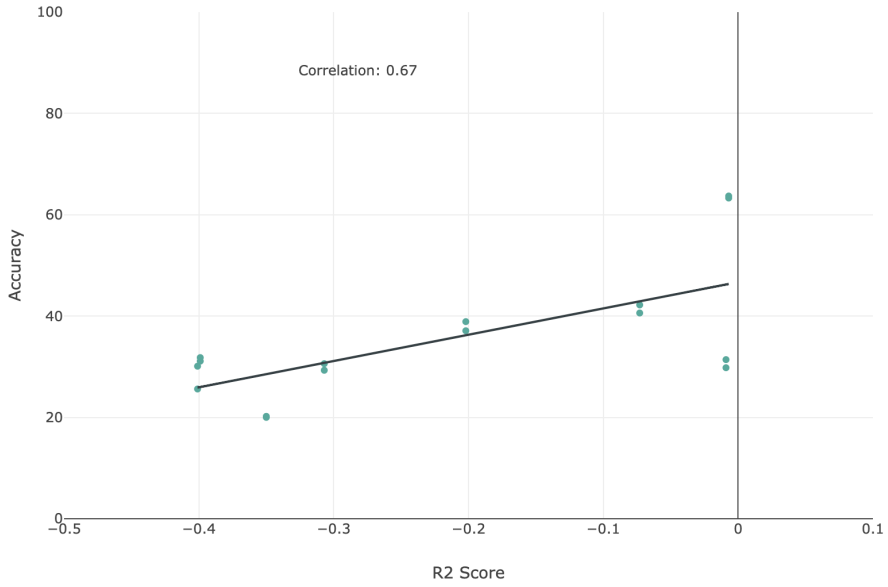


Figure 30: Score Correlation for N2M2P Models

The 4 best models of the N2M2P task, based on the average placement, is used to compute confidence intervals. Confidence intervals for multiple confidence levels are displayed in Table 39.

Table 39: Confidence Intervals for the 4 Best N2M2P Models

Confidence Level	Acc Avg	Acc Lower	Acc Upper	R2 Avg	R2 Lower	R2 Upper
90%	0.525	0.374	0.675	–0.040	–0.084	0.005
95%	0.525	0.321	0.728	–0.040	–0.100	0.020
99%	0.525	0.151	0.898	–0.040	–0.150	0.071

5.5.5 Notes to MLPFs and Pianists (N2MP)

Selection of N2MP models for retraining is described in subsection 5.4.4. The accuracy scores of the retrained N2MP models are presented in Table 40. The models are ordered by the average placement (*Avg No.*) in the prediction task (R^2 *No.*) and in the classification task (*Acc No.*).

Table 40: N2MP Results

RNN	Selection	Deviation	Save	R^2	Acc	R^2 No.	Acc No.	Avg No.
LSTM	Comb	Average	R^2	–0.323	75.3%	12	6	9
LSTM	Comb	Average	Acc	–0.309	75.3%	11	7	9

GRU	Acc	Average	End	–0.042	67.8%	2	16	9
LSTM	Comb	Average	End	–0.293	72.9%	10	9	9.5
GRU	Acc	Average	Acc	–0.081	70.2%	8	12	10
LSTM	R^2	Average	R^2	–1.288	79.2%	22	3	12.5
GRU	R^2	Score	R^2	–0.476	70.8%	14	11	12.5
LSTM	Comb	Score	R^2	–0.641	72.7%	16	10	13
LSTM	R^2	Average	End	–1.488	77.0%	23	4	13.5
GRU	Acc	Average	R^2	–0.054	64.6%	4	24	14
LSTM	R^2	Average	Acc	–1.707	83.1%	28	1	14.5
LSTM	Acc	Average	Acc	–0.052	62.0%	3	27	15
GRU	R^2	Score	Acc	–0.714	69.1%	17	14	15.5
GRU	Acc	Score	Acc	–0.592	65.2%	15	20	17.5
LSTM	Acc	Average	R^2	–0.014	55.6%	1	34	17.5
GRU	R^2	Score	End	–0.855	68.0%	21	15	18
GRU	Comb	Average	End	–0.064	60.5%	5	31	18
LSTM	Comb	Score	End	–0.820	65.2%	18	19	18.5
GRU	R^2	Average	Acc	–5.730	80.6%	36	2	19
GRU	Comb	Average	Acc	–0.064	60.5%	6	32	19
GRU	R^2	Average	End	–4.550	76.4%	34	5	19.5
LSTM	Comb	Score	Acc	–0.820	65.2%	19	21	20
GRU	Comb	Average	R^2	–0.070	56.6%	7	33	20
GRU	R^2	Average	R^2	–5.341	74.2%	35	8	21.5
GRU	Comb	Score	Acc	–1.652	67.1%	27	17	22
LSTM	R^2	Score	Acc	–2.596	69.5%	32	13	22.5
GRU	Acc	Score	End	–0.833	63.2%	20	25	22.5
LSTM	Acc	Average	End	–0.166	52.3%	9	36	22.5
GRU	Comb	Score	End	–1.509	64.9%	24	22	23
GRU	Acc	Score	R^2	–0.324	55.3%	13	35	24
LSTM	R^2	Score	End	–2.320	66.9%	31	18	24.5
LSTM	Acc	Score	Acc	–1.534	62.4%	25	26	25.5
LSTM	R^2	Score	R^2	–2.616	64.7%	33	23	28
GRU	Comb	Score	R^2	–1.641	61.0%	26	30	28
LSTM	Acc	Score	End	–1.903	61.2%	29	28	28.5
LSTM	Acc	Score	R^2	–1.903	61.2%	30	29	29.5

Figure 31 shows the correlation between the R^2 scores on the prediction task and the accuracy score on the classification task. The correlation between the scores is -0.45.

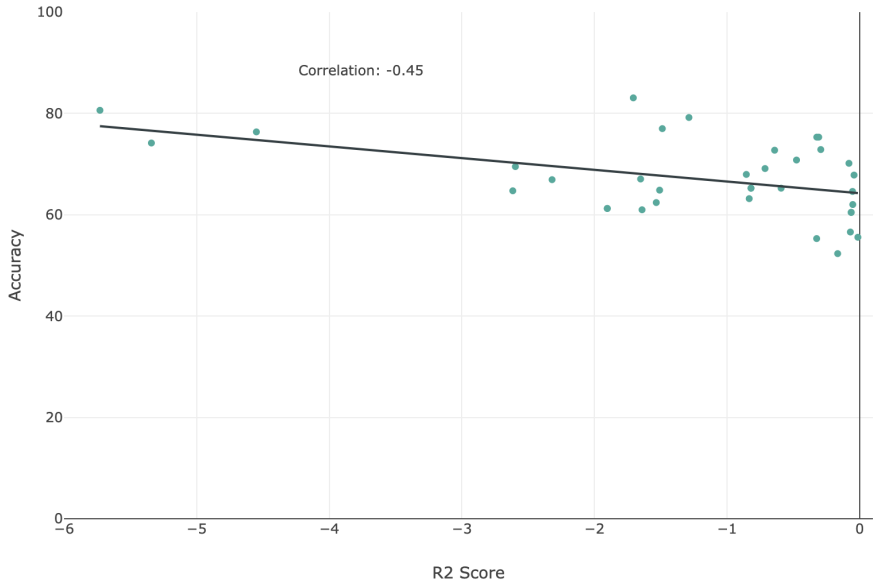


Figure 31: Score Correlation for N2MP Models

Table 41: Confidence Intervals for the 4 Best N2MP Models

Confidence Level	Acc Avg	Acc Lower	Acc Upper	R2 Avg	R2 Lower	R2 Upper
90%	0.728	0.687	0.770	–0.242	–0.399	–0.085
95%	0.728	0.672	0.785	–0.242	–0.455	–0.029
99%	0.728	0.625	0.832	–0.242	–0.632	0.148

6 Discussion

This chapter presents discussions of model performance on each task, as well as data reliability and processing. The Mid-Level Perceptual Features (MLPF) dataset is analyzed in section 6.1 and is considered to be unreliable. Section 6.2 discusses the processing methods that were used. The ground truth calculation method, based on Kernel Density Estimation (KDE), is analyzed. The results of reproducing the KDE model is discussed in section 6.3. The next sections include the performance of the models on each task, and relate it to the researtch questions. Section 6.1 and section 6.5 are about the MLPF prediction task and pianist recognition task, respectively.

6.1 Mid-Level Perceptual Features (MLPF) Dataset

The dimensions of the MLPF dataset had poor reliability, even when the labels were averaged. The ICCs of the dimensions in Table 13 show that the labels were not reliable on their own. The largest ICC in the case of single score (ICC(2,1)) was 0.35, for the Bright/Dark- and Optimistic/Pessimistic dimensions. However, the lower bounds of the 95% confidence intervals were 0.24 and 0.22 respectively. These values were not close to the score of 0.5, which Koo and Li used as the dividing value for poor and moderate reliabilty [44].

In the case of the average measure (ICC(2,k)), most of the MLPFs were still considered to have poor reliability. According to Koo and Li [44], none of the dimensions were considered to have good reliability. This is achieved when the lower boundary of the 95% confidence interval is between 0.75 and 0.9. Only 3 of the 25 dimensions were considered moderately reliable: Saturated/sparse pedaling, bright/dark tone and optimistic/pessimistic emotion. The rest were considered to have poor reliability. The least reliable dimensions regarding average measure were stable/unstable beat, subtle/obvious change of pedal and harmonious/disproportioned music making. These dimensions had lower bounds of 0.07, 0.07 and 0.08 respectively.

The results from the ICC analysis indicated that it would be difficult to train a model to predict the MLPF features. First, the data must have been aggregated to be considered more reliable. This reduced the amount of training data from the total amount of responses to the amount of distinct musical segments. In the case of the MLPF dataset, the amount of data samples was then reduced from 1690 to 115. This amount was too small to expect good performances by the models. Secondly, most of the dimensions werre considered to have poor reliability. Even though some of the dimensions would be possible to predict, the unreliable dimensions could decrease the overall score of all the models on the prediction task.

The MLPF dataset was not completed at the time of writing. The reliability of the dataset should therefore be re-evaluated when all the data is collected. Table 12 shows that only a three users have labelled the whole dataset. This was an issue, since it made it difficult to compute the ICC of the whole dataset. Conclusions based on the reliability of four experts cannot be given too much

weight. However, the results indicated that the MLPFs were subjective in nature. This finding coincides with the findings of Aljanaki and Soleymani [1].

6.2 Processing

The processing of piano performances was conducted in several steps. Each of the steps increased the complexity and could potentially have introduced errors. The alignment process of notes was necessary for individual performances to be compared to each other, when the data was in midi format. Table 15 shows significant differences in the amount of aligned notes. The notes of the score were correctly aligned 91.5% of the time. This was suprisingly low since the score is aligned with itself. Johannson was the performer with most aligned notes, 82.7% of his notes were correctly aligned. Mordvinov had the least amount of correctly aligned notes, with 66.8%.

The computation of deviations from the average was dependent on a significant amount of performances. Because of alignment errors, each average note was computed based on a subset of the performances. Table 17 and Table 16 shows the amount of notes that were used for each average note, for the pianist and MLPF datasets. In the MLPF case, all performers were used to compute the average for 93.4% of the notes. For the pianist dataset, this was only true in 71.1% of the cases. When the average notes were computed based on only a few pianists, they were less reliable. In these cases, deviations from the score might have been a better option.

The apex method for aggregating labels was affected by the configuration of the kernel density estimation. This included the bandwidth parameter, the kernel and the amount of samples to compute the distribution. The small size of the MLPF dataset made it possible to go through the whole dataset and evaluate the appropriateness of the method. The default settings were considered to work well for the given task. In the case of mean and median, there were no parameters to configure. The choice of using mean or median would not introduce any bias. However, the methods were not chosen because of their inability to model the collective opinion of the domain experts. Figure 12 illustrates this point.

6.3 Reproduced Kernel Density Estimation (KDE) Model

This section discusses the reproducing of the KDE method on the original k-fold pianist recognition task. Original and optimized parameters are analyzed in subsection 6.3.1 and subsection 6.3.2 respectively.

6.3.1 Original parameters

The original version of the KDE method did not perform as well as in the original paper. The score of the reproduced model (F1 score: 0.458) was significantly lower than the score of the orginal method (F1 score: 0.807)[24]. The reproduced model showed a clear bias towards classifying the unknown pianist as Abdelmoula. He was predicted to be the unknown pianist in 35 of 72 cases, even though the nine pianists were equally represented in the test- and training data. Figure 18 shows the reason for the bias. Even though the known pianist distribution of Abdelmoula closely resembled the unknown pianist distribution of himself, the computed divergence was infinite. The KDE model predicted the pianist to be the one with the minimal summed divergence. As in the case of Figure 18, all of the pianists had have infinite summed entropies. This was regardless of the entropies of the other dimensions or the weighting between the dimensions. Since all the summed entropies were equal, the minimum was considered to be the first value. Abdelmoula was therefore predicted to be the correct pianist in all the cases where the divergence was infinite in at least one feature dimension for each pianist.

A computed divergence of infinite entailed that the distributions were too dissimilar to compare. The test- and training distributions of Abdelmoula in Figure 18 seemed comparable, even though the divergence was infinte. This indicated that the chosen bandwidth was too small. This did not mean that the chosen parameter was too small in the original paper. Even though the input data was the same, the feature scaling might have differed between the two versions of the method. The reason for this difference was searched for, but without any results. Therefore, the optimized version of the reproduced KDE method was considered to be a better candidate for a fair comparison.

6.3.2 Optimized Parameters

The paper of Rafee et al. [24] stated that their values for bandwidths was found by optimization, but the process was not described. In the reproduced case, the training folds were used as training data and the test fold was used to tune the parameters. This approach regarded the test data as validation data, and was not a good measure of the method's ability to generalize. However, it was seen as the most appropriate approach when comparing with the original method.

All of the scores for single input features were greater than what was stated by Rafee et al. [24] in the original paper. The most notable difference was the method's ability to perfectly predict all the pianists based on only the onset time feature. Figure 18 shows the known pianist distributions from fold 2 to

8 (dark green for Abdelmoula and blue for other pianists), and the test distribution of Abdelmoula from the 1st fold (light green). The distributions were quite different, which made sense. All the pianists played the notes at their own tempo. The onset time was computed by subtracting the pianist's onset time from the onset time of the average performer. That means that the feature was centered, but not standardized. This had the effect that later notes had accumulated larger deviations to the average notes, which made it easier to predict pianists based on its distributions. The predictive power of the KDE method was also enhanced by randomly dividing notes into folds, as if the notes were independent of each other. The accumulated deviations of onset times was distributed equally to all the folds, which means that the test distributions would be similar to the training distribution. This was not the case for the other features. Note duration, velocity, inter onset interval or offset time duration did not accumulate like onset time. This can be observed by studying the distribution differences between onset time (Figure 19) and the rest of the features (Figure 20, Figure 21, Figure 22, Figure 23).

6.4 MLPF Prediction Task

The performance of the machine learning models on the prediction task were poor, even though they generally have higher R^2 scores than the baseline models and domain experts. As can be observed in section 4.4, all of the machine learning models had negative R^2 scores on the test set of the MLPF dataset. This is typically not observed in prediction tasks, where simple models should easily fit the data better than the average of that data. However, there were some challenges with the given MLPF prediction task that made it more difficult: First, the MLPF data was unreliable. As mentioned in section 6.1, only 3 of the 25 dimensions were considered to be moderately reliable. The rest were considered to have poor reliability. Secondly, the MLPF dataset was small. The aggregated MLPF data consisted of 115 samples, which was split into training-, validation- and test data. This was to small to produce generalizable models and to be able to test them fairly. Thirdly, the input data consisted of sets of subsequent notes. This datatype is difficult to work with, where both note alignment- and feature extraction tasks introduce challenges. Finally, the relationship between the input data and the output data is complex. As stated in section 1.1, mid-level features are difficult to define clearly. Complex relationships entail complex models with large amounts of parameters. Larger amounts of parameters requires more data for training.

6.4.1 Baseline

The scores of the *Center* method (Table 21) shows that the data samples in the test and validation sets were less centralized than the data in the training set. This means that validation and test sets consisted of more distinct styles on average than the training set. *Training Average* had higher scores for all subsets, in relation to the *Center* basline model. However, both baseline models had mostly negative R^2 scores.

6.4.2 Human

The average human domain experts had lower R^2 score than the baseline models on the MLPF dataset. Four of the human experts were performing equally or better than the *Center* method on all the data. This was observed by comparing the R^2 scores of the humans in Table 22 to the R^2 scores for the baseline models in Table 21. No humans scored better than the *Training Average* method on the validation data, and only one human scored better than *Training Average* on the test data. The fact that the baseline methods were simple in nature and disregarded the input values when making predictions, while having better scores than the domain experts, was not promising for the task. The R^2 scores for the experts were also low. Only the top two experts (109, 86) were able to achieve positive scores for the whole or subsets of the dataset. Positive scores mean that they are able to better predict the values than the average value for all the dimensions in that particular data set. User 109 ranked 1st on all- ,

training- and test data, and 6th on the validation data. Even though user 109's performance was the best in the pool of domain experts, its R^2 scores were low.

Table 23 shows large differences between the boundaries of the confidence intervals. This was due to the large deviation of human scores. The upper boundary of the averaged human R^2 on the whole dataset at the 99% level was -0.235. Even at the upper boundary at the 99% confidence level, the average human scored worse than the *Center* method. As shown in Table 33, the *Center* method scored -0.17 on the whole dataset.

The human domain experts had an advantage over other methods when it came to predicting the ground truth labels. That is because the ground truth labels were based on the labels of the experts. However, the relationships between the participants' labels and the ground truth labels are complex. The method of computing the ground truth labels was based on finding the x-values of the apexes of the label distributions. One of the disadvantages for the humans is that they could only label the segments of music on the likert scale from 1 to 7. The baseline- and machine learning models were able to output real numbers. This is a major disadvantage when it comes to performance, because the humans would most likely introduce errors for each dimension and never be able to predict perfectly. At the same time, introducing a discrete scale with more options or continuous scale might have lead to more inconsistencies and confusion for the experts.

6.4.3 Notes-to-MLPFs (N2M)

For the training- and validation sets, all the models had better R^2 scores the human and the baseline models. For the test set, three of eight models score better than the *Training Average* baseline model. All of the models had worse scores than one domain expert (User 109) on the test set. However, all of the N2M models scored better than the rest of the humans. The N2M models that used deviations from the average, had better R^2 scores than the average human at the 95% confidence level.

The results in Table 24 show a decrease in R^2 scores based on the subset of the MLPF dataset: There were significant decreases in R^2 scores from the training set to the validation set, and from the validation set to the test set. The decreases in scores indicated overfitted models, which were unable to generalize on new data. This was related to the size of the MLPF dataset, which was to small for generalizing. All of the scores for the test set were negative. Negative scores entail that the models are worse at predicting the values of the test set than the average of the test set.

Table 26 shows that at the 95% confidence level, the upper boundary of *Score* was smaller then the lower boundary of *Average*. This indicated that the use of deviations from the average performance performed better than the deviations from the score. Table 25 shows that the confidence intervals of LSTM and GRU overlapped. Based on this, the higher average score of LSTM could have been a result of chance. Only four datapoints were used to calculate the confidence intervals. Even though the student's t-distribution takes the sample

size into account, the confidence interval would be more reliable with larger sample sizes. Additionally, the average scores should not be averaged across configuration combinations. This was done to have more datapoints for interval computation.

6.4.4 Notes-to-MLPFs-and-Pianists (N2MP)

The models using the N2MP architecture performed similarly to the models to the N2M architecture. A decrease in R^2 scores was observable, from the training set to the validation set and from the validation set to the test set. All the test scores of the N2MP models were negative. This entails that the models were worse than the average of the test set. The GRU and Average configuration scored better than the average human at the 99% level. This was also true for LSTM and Average configuration at the 90% level of confidence.

Table 31 shows that the models using deviations from the average performance had better R^2 scores than the models that used deviations from the score, on average, at the 95% confidence level. Both in the case of LSTM and GRU were the lower boundaries for *Average* larger than the upper boundaries for *Score*. Table 30 indicate that GRU scores better than LSTM, because of higher averaged scores and interval boundaries. Yet, the difference could not be observed for all configurations at the 90% confidence level. In the case of deviations from the average performance, the lower interval boundary of GRU was larger than the upper boundary of LSTM. This was not the case when deviations from the score was used.

6.4.5 Best Method of MLPF Extraction (RQ1)

Both the N2M- and the N2MP architectures produced only models that had negative R^2 scores on the MLPF prediction task. These results were worse than predicting the average of the test set. The models trained with the N2M- and N2MP architecture generally have higher scores than the baseline methods and the human domain experts. The best N2MP model, on average, was using GRU and deviations from the average performance. This model had a confidence interval from -0.07 to -0.05 at the 90% level of confidence. In the N2M case, the best models used deviations from the average. These N2M models had an confidence interval from -0.18 to 0.035 at the 90% level. The interval of the human domain experts was from -0.825 to -0.381 at the same level of confidence. It was not possible to claim that the average human scored better than the *Center* method at this level of confidence. Even though the *Center* model had an R^2 score of -0.67 on the test set, the lower bound of the average human was lower. The average human score was worse than the *Training Average* baseline model, which scored -0.13 on the test set.

The poor results on the prediction task was linked to the reliability discussed in section 6.1. When 22 of 25 dimensions was considered to have poor reliability, the models would not be able to learn the patterns of the data. Additionally, the poor results was also linked to the size of the MLPF dataset. Both the N2M

and the N2MP architectures had large decreases in R^2 scores from the training set to the validation set, and from the validation set to the test set. This indicated overfitted models. As mentioned in section 6.1, the MLPF dataset had 115 samples after processing. The training set of 92 samples was not likely to produce generalizable models. The validation and test sizes of 11 and 12, respectively, were too small to be able to fairly evaluate the models.

The results showed that models that used deviations from the average performance scored better than the models that used deviations from the score, on average. This was shown for both the N2M and the N2MP architectures at the confidence level of 90%. The results also showed that LSTM performed better, on average, when using N2M. GRU performed better, on average, in the case of N2MP. Neither of the results were significant at the 90% level of confidence. It was therefore not possible to conclude which of the RNN cells were best for extracting MLPFs.

6.5 Pianist Recognition Task

This section includes the comparison of models on the pianist recognition task. The baseline model, KDE models and N2P models are compared to the models which use the MLPFs as intermediate steps: N2M2P and N2MP. The The cost of explainability is computed in subsection 6.5.6, by analyzing the accuracy differences of N2M2P and N2MP in relation to N2P. Finally, the results are related to **RQ2** in subsection 6.5.7.

6.5.1 Baseline

The baseline model disregards the input when classifying the pianist. The fact that the model outputs the same prediction every time reduced its usefullness. The model was used as a benchmark for the rest of the models, even though it produced results that were far worse.

6.5.2 KDE

Table 34 shows that there are small differences between the original and the optimized models when they were evaluated on the pianist classification task described in section 4.5. Both of the versions had more than 6 times higher accuracy than the baseline model on the test set.

The differences between the original and the optimized models were larger when they were evaluated based on the original classification task (described in subsection 4.3.5), using k-fold and no chunking. This was an unexpected result since the original method should have been optimized for the original task, and was unlikely to improve performance when tested on a different task. This indicated that the original classification task was wrongly reproduced.

6.5.3 Notes-to-Pianists (N2P)

Table 35 shows that the N2P architecture performed well on the pianist classification task. All of the models using deviations from the average have better accuracies than the KDE models. This is also true in the average case, where the N2P models perform better than the KDE models at the 95% level of confidence. The good scores on the test set showed that the approach of using RNNs to distinguish pianists was appropriate.

The confidence intervals presented in Table 36 and Table 37 substantiated the discussion in subsection 6.4.5. Models that used deviations from the average performer performs better than the models that use of deviations from the score, on average. This was determined at the 95% confidence level.

6.5.4 Notes-to-MLPFs-to-Pianists (N2M2P)

Figure 30 shows that there was a positive correlation (0.67) between the R^2 scores on the prediction task and the accuracy score on the classification task for the N2M2P models. Additionally, Table 38 shows that the two best models

on the classification task used the regressor model which performed the best on the prediction task. There was a significant decrease in recognition accuracy to the next model: From 63.3% to 42.2%. These results indicated that a better ability to extract the MLPFs entailed better results on the classification task, when using this architecture.

When computing the average scores of the N2M2P models, only the best performing models were interesting to look at. This was because a poor model would not be considered for the given task. At the same time, the sample size should not have been too small either. Subsection 6.4.5 indicated that deviations from the average produced better results than the deviations from the score. N2M2P models were therefore only included if they used deviations from the average. It was important too rank the models based on both the tasks. In order to answer the second research question, models must be able to predict the MLPF when they were evaluated by accuracy. Even though the predictions were poor in an absolute sense, which was discussed in subsection 6.4.5, the relative comparisons of the models were interesting for the second research question.

The relative ranking method of using the average placement on each task was not a perfect solution. The method could be unfair in situations where there were small absolute differences between sets of models. At the same time, using absolute ranking methods could treat models unfairly in cases of outliers. In the case of the N2M2P methods, the ranking worked well. Figure 30 shows that both the scores correlate. This entailed that the combined ranking did not contradict both of the individual rankings to significant extents.

6.5.5 Notes-to-MLPFs-and-Pianists (N2MP)

Figure 31 shows that there was a negative correlation (-0.45) between the R^2 scores on the prediction task and the accuracy score on the classification task for the N2MP models. These results indicated that the model regressors optimized for other features than the MLPFs to perform better on the classification task. Models using the N2MP architecture are able to specialize on either of the tasks. This is not possible in the seperate training loops of the N2M models.

The same ranking method was used in the N2MP case as is described in subsection 6.5.4. In the N2MP case, the correlation of the R^2 and Accuracy scores was negative. This made it difficult to rank the models fairly, since the individual rankings were likely to be inverse of each other. However, the method of ranking seemed to be working well. Top ranked N2MP models had relatively high scores on both tasks compared to the other N2MP models.

6.5.6 Cost of Explainability

The best N2M2P model had an accuracy score of 63.7% and an R^2 score of -0.007. In the N2MP case, the best model had an accuracy score of 75.3% and an R^2 score of -0.323. These models was compared to the best N2P model, which had an accuracy score of 82.8%. There was a decrease in accuracy when introducing the MLPFs. The reductions were 7.5% and 19.1% for N2MP and

N2M2P, respectively. A decrease in accuracy score was also observable when improving the predictive ability of MLPFs. The improvement in R^2 score from -0.323 (N2MP) to -0.007 (N2M2P) reduced the accuracy by 11.3 %.

The average scores and confidence intervals for the 4 best models of each architecture were computed. The average accuracies were 80.2%, 52.5% and 72.8% for N2P, N2M2P and N2MP, respectively. In this case, the explainability costs from N2P were 27.7% to N2M2P and 7.4% to N2MP. The improvement in R^2 score from -0.242 (N2MP) to -0.040 (N2M2P) reduced the accuracy by 20.3 %. The confidence intervals were analysed in order to see if the averaged results could be attributed to chance or not. The differences in average accuracy scores between N2P and N2M2P was determined at the 95% confidence level. In the case of N2MP, the results showed that the architecture was outperformed by the N2P architecture at the 90% confidence level. When it came to the differences in R^2 score between the N2M2P- and N2MP architectures, the difference was significant at the 90% level of confidence.

6.5.7 Impact of MLPFs as an Intermediate Step (RQ2)

The overall impact of introducing MLPFs as an intermediate step in the recognition task was negative. That is because the explainability of the MLPFs was unreliable and the decrease in accuracy was significant. Subsection 6.4.5 showed that none of the models were able to accurately predict the MLPFs. Humans cannot base their understanding on the MLPFs if the values for each MLPF is likely to be wrong. Furthermore, the MLPFs do not represent something meaningful if they are subjective by nature.

Even though the predicted values for each MLPF might be wrong, the introduction of MLPFs can improve explainability. Models which use MLPFs as an intermediate step will base their pianist recognition on inaccurate values, since the MLPFs are not predicted correctly. Such models will convey the inaccurate assumptions of the MLPFs to the user. A domain expert will be able to evaluate the values for each MLPF and come to his or her own conclusion. The output of the regressor model can be altered, to align with the views of the domain expert. Then, the values can be given to the classifier model for pianist recognition and possibly produce better results. The domain expert will be able to learn from the classifier part of the model, and connect performance styles to each pianist. The explainable MLPFs can also be used to pinpoint the weaknesses of models. However, the knowledge cannot be fully trusted since the classifier model is trained on inaccurate MLPFs.

The decreases presented in subsection 6.5.6 showed that there were large decreases in accuracy when introducing MLPFs as an intermediate step. For the best models of the N2MP- and N2M2P architectures, the decreases were 7.5% and 19.1%, respecitvely. It is difficult to compare these results to the results of Chowdhury et al., since they used correlation coeffiecient as the score metric. The architectures A2mid2E (N2M2P) and A2Mid2E-Joint (N2MP) were compared to the A2E (N2M): The decreases were 7% in the case of A2midE and 1.5% for A2Mid2E-Joint [2]. Correlation coefficient has a bigger range than

accuracy, so it can be argued that the cost of explainability in this case is larger than in the paper of Chowdhury et al.

7 Conclusion

The small size and poor reliability of the MLPFs dataset made it challenging to answer the research questions of this thesis. Only 3 of the 25 MLPFs used for model training were considered (moderately) reliable: Saturated/sparse pedaling, bright/dark tone, and optimistic/pessimistic emotion. This reliability was only achieved when the human labels were averaged. Averaging of labels entailed reducing the number of samples from 1690 to 115. Such a small dataset, with mostly unreliable features, was not considered likely to produce models that could predict MLPFs accurately.

The first research question is about the best methods for extracting MLPFs. All the models had negative R^2 scores, which means that the models were worse at predicting the MLPFs than the average of the test set. The domain experts had, on average, worse scores than the N2M models that used deviations from the average. That was determined at the 95% confidence level. The N2MP models that used deviations from the average performance and GRU performed better than the average human domain expert at the 99% confidence level. One expert was able to achieve a positive R^2 score on the test set of the MLPF dataset. Thus, the expert outperformed all the models. It was shown that the deviations from the average performance were better at predicting the MLPFs than the deviations from the score. That was determined at the 95% level of confidence for both the N2M and N2MP architectures. The results did not indicate which RNN cell was better for MLPF extraction. For N2M, it was indicated to be the LSTM cell. For N2MP, it was indicated to be the GRU cell. The results regarding the best RNN cell were not significant at the 90% confidence level.

The second research question is about the impact of introducing MLPFs when recognizing individual pianists. Introducing MLPFs as an intermediate step entailed significant decreases in accuracy. For the best models of the N2MP- and N2M2P architectures, the decreases were 7.5% and 19.1%, respectively. The accuracies of these architectures were compared to the N2P architecture. The best N2P model was able to recognize the pianists with an accuracy of 82.8%. That was higher than the reproduced model that used Kernel Density Estimation (KDE), which scored 74.5% on the same task. Because of the poor results of MLPFs prediction linked to the first research question, explainability via MLPFs would not be considered trustworthy.

8 Future Work

In order to research musical concepts with data-driven approaches, reliable datasets need to be collected. Aljanaki and Soleymani described the mid-level features to be subjective and difficult to define clearly [1]. However, it is not scientific to use such features if they cannot be defined or objectively measured. As long as the data for these features are unreliable, different approaches cannot be researched in a meaningful way. Future work can therefore collect more data for the given set of mid-level perceptual features (MLPFs) or new data for new sets of features. This thesis was based on a partially completed dataset. Hopefully, the completed dataset will result in more reliable features than shown in this thesis.

Data reliability can be an issue for pianist classification tasks as well. It was concluded that deviations from the average performance better capture the individuality of a performer than deviations from the score. Eleven pianists have been used in this thesis, in contrast to the 9 in the paper of Rafee et al. [24]. These amounts of performers can be hypothesized to produce good averages for model training. However, this hypothesis has not been researched. Future work can address the issue of how many pianists are needed to produce a reliable average for deviation computation. In the cases where average performance is computed, the pianists' level can also be researched.

Only one piece of Schubert is used for classification in the paper of Rafee et al. [24] and this thesis. This piece was chosen because of the many pianists who have played it to compute a reliable average performance. However, methods to be developed should be able to generalize across pieces, performers and epoques. Therefore, generalizable models should be addressed in future work. That can entail the research of different feature extraction methods and machine learning architectures.

A competition of predicting MLPFs has been planned at SNU. The goal is to make more people interested in music and artificial intelligence. Then, hopefully, new approaches will be developed for the MLPF prediction task, which will produce better results.

References

[1] Anna Aljanaki and Mohammad Soleymani. A data-driven approach to mid-level perceptual musical feature modeling, 2018.

[2] Shreyan Chowdhury, Andreu Vall, Verena Haunschmid, and Gerhard Widmer. Towards explainable music emotion recognition: The route via mid-level features, 2019.

[3] Efstathios Stamatatos. Quantifying the differences between music performers: Score vs. norm. In *ICMC*, 2002.

[4] Terry B. Ewell. Music fundamentals 1: Pitch and major scales and keys. `http://cnx.org/contents/88c940cd-cf98-4831-91a8-4fc753575bb6@1.10`, 2013.

[5] Roger B. Dannenberg. An introduction to music concepts. `https://www.cs.cmu.edu/~music/cmsip/readings/music-theory.htm`.

[6] Roger B. Dannenberg. Style in Music. In *The Structure of Style*, page 45. 2010.

[7] Robert DiPietro and Gregory D. Hager. Chapter 21 - deep learning: Rnns and lstm. In S. Kevin Zhou, Daniel Rueckert, and Gabor Fichtinger, editors, *Handbook of Medical Image Computing and Computer Assisted Intervention*, The Elsevier and MICCAI Society Book Series, pages 503–519. Academic Press, 2020.

[8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[9] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[10] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[11] Nikolaus Hansen and Andreas Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2):159–195, 06 2001.

[12] Anders Kofod-Petersen. How to do a structured literature review in computer science. `https://research.idi.ntnu.no/aimasters/files/SLR_HowTo2018.pdf`, 2018.

[13] Claes Wohlin. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *EASE '14*, 2014.

[14] Dasaem Jeong, Taegyun Kwon, Yoojin Kim, and Juhan Nam. Graph neural network for music score data and modeling expressive piano performance. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3060–3070. PMLR, 09–15 Jun 2019.

[15] Eita Nakamura, Kazuyoshi Yoshii, and Haruhiro Katayose. Performance error detection and post-processing for fast and accurate symbolic music alignment. In *ISMIR*, 2017.

[16] Beici Liang, György Fazekas, and Mark Sandler. Measurement, recognition, and visualization of piano pedaling gestures and techniques. *Journal of the Audio Engineering Society*, 66:448–456, 06 2018.

[17] Carlos Cancino-Chacón and Maarten Grachten. A computational study of the role of tonal tension in expressive piano performance, 2018.

[18] Elaine Chew. *Towards a mathematical model of tonality*. PhD thesis, Massachusetts Institute of Technology, 2000.

[19] Elaine Chew. Playing with the edge: Tipping points and the role of tonality. *Music Perception: An Interdisciplinary Journal*, 33(3):344–366, 2016.

[20] Carlos Cancino-Chacón, Silvan Peter, Shreyan Chowdhury, Anna Aljanaki, and Gerhard Widmer. On the characterization of expressive performance in classical music: First results of the con espressione game, 2020.

[21] Cynthia Liem and Alan Hanjalic. Expressive timing from cross-performance and audio-based alignment patterns: An extended case study. pages 519–524, 01 2011.

[22] Maarten Grachten, Carlos Eduardo Cancino-Chacón, Thassilo Gadermaier, and Gerhard Widmer. Towards computer-assisted understanding of dynamics in symphonic music, 2016.

[23] Carlos Eduardo Cancino-Chacón, Thassilo Gadermaier, Gerhard Widmer, and Maarten Grachten. An evaluation of linear and non-linear models of expressive dynamics in classical piano and symphonic music. *Machine Learning*, 106:887 – 909, 2017.

[24] Syed Rifat Mahmud Rafee, Gyorgy Fazekas, and Geraint A. Wiggins. Performer identification from symbolic representation of music using statistical models, 2021.

[25] Zhengshan Shi. Computational analysis and modeling of expressive timing in chopin mazurkas. In *ISMIR*, 2021.

[26] Akira Maezawa, Kazuhiko Yamamoto, and Takuya Fujishima. Rendering music performance with interpretation variations using conditional variational rnn. In *ISMIR*, 2019.

[27] Hendrik Schreiber, Frank Zalkow, and Meinard Müller. Modeling and estimating local tempo: a case study on chopin's mazurkas. 10 2020.

[28] Shreyan Chowdhury and Gerhard Widmer. Towards explaining expressive qualities in piano recordings: Transfer of explanatory features via acoustic domain adaptation, 2021.

[29] Hsiao-Tzu Hung, Joann Ching, Seungheon Doh, Nabin Kim, Juhan Nam, and Yi-Hsuan Yang. Emopia: A multi-modal pop piano dataset for emotion recognition and emotion-based music generation, 2021.

[30] TJ Tsai and Kevin Ji. Composer style classification of piano sheet music images using language model pretraining, 2020.

[31] Gissel Velarde, Tillman Weyde, Carlos Eduardo Cancino-Chacón, David Meredith, and Maarten Grachten. Composer recognition based on 2d-filtered piano-rolls. In *ISMIR*, 2016.

[32] Yu-Siang Huang and Yi-Hsuan Yang. Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions, 2020.

[33] Sangeetha Rajesh and N J Nalini. Musical instrument emotion recognition using deep recurrent neural network. *Procedia Computer Science*, 167:16–25, 2020. International Conference on Computational Intelligence and Data Science.

[34] Gino Brunner, Andres Konrad, Yuyi Wang, and Roger Wattenhofer. Midi-vae: Modeling dynamics and instrumentation of music with applications to style transfer, 2018.

[35] Changfeng Chen and Qiang Li. A multimodal music emotion classification method based on multifeature combined network classifier. *Mathematical Problems in Engineering*, 2020:1–11, 08 2020.

[36] Nadine Hajj, Maurice Filo, and Mariette Awad. Automated composer recognition for multi-voice piano compositions using rhythmic features, n-grams and modified cortical algorithms. *Complex Intelligent Systems*, 08 2017.

[37] Christof Weiss and Meinard Müller. Tonal complexity features for style classification of classical music. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 688–692, 2015.

[38] Yi-Hui Chou, I-Chun Chen, Chin-Jui Chang, Joann Ching, and Yi-Hsuan Yang. Midibert-piano: Large-scale pre-training for symbolic music understanding, 2021.

[39] Shreyan Chowdhury and Gerhard Widmer. On perceived emotion in expressive piano performance: Further experimental evidence for the relevance of mid-level perceptual features, 2021.

[40] Shreyan Chowdhury, Verena Praher, and Gerhard Widmer. Tracing back music emotion predictions to sound sources and intuitive perceptual qualities, 2021.

[41] A. Braginsky and Y. Corporation. International piano-e-competition. `https://www.piano-e-competition.com/`, 2002. Accessed on May 26th 2022.

[42] B.A. Therkelsen. Mid-level perceptual feature dataset. `https://github.com/batherk/performance-style/blob/thesis-delivery/data/raw/labelling/total.csv`, 2022. Private repository. Available on request.

[43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[44] Terry K. Koo and Mae Y. Li. A guideline of selecting and reporting intraclass correlation coefficients for reliability research. *Journal of chiropractic medicine*, 15(2):155–163, Jun 2016. 27330520[pmid].