Identification of expressive descriptors for style extraction in music analysis using linear and nonlinear models

Mauro Alejandro Jimenez Medina

November 18, 2022

UNIVERSIDAD DEL NORTE

Identification of expressive descriptors for style extraction in music analysis using linear and nonlinear models

Presented by:

Mauro Alejandro Jimenez Medina

Supervisor:

Winston Spencer Percybrooks, Ph.D.

Department of Electrical and Electronics Engineering

Biomedical Signal Processing and Artificial Intelligence Laboratory

November 18, 2022

Mauro Alejandro Jimenez Medina
Identification of expressive descriptors for style extraction in music analysis using linear and nonlinear models
November 18, 2022
Supervisor: Winston Spencer Percybrooks, Ph.D. **UNIVERSIDAD DEL NORTE**
Biomedical Signal Processing and Artificial Intelligence Laboratory
Department of Electrical and Electronics Engineering
Barranquilla

Advisor Revision

I, the undersigned, declare that the work here is original and it has been completed only with the help of the references mentioned.

Barranquilla, November 18, 2022

Winston Spencer Percybrooks,
Ph.D.

Abstract

Formalization about expressive interpretations is still considered relevant due to the complexity of music. Expressive interpretation forms an important aspect of music, taking into consideration different conventions like genres or styles that a performance can develop over time.

Modeling the relationship between musical expressions and structural aspects of the acoustic information requires a minimum probabilistic and statistical foundation for robustness, validation and reproducibility of computational applications. Therefore, a cohesive relation and justification about the results is necessary.

This thesis is supported by the theory and applications of discriminative and generative models within the framework of machine learning and the relationship of systematic procedures with musicology concepts using data mining.

Results were validated using statistics tests and a non-parametric experimentation by implementing a set of metrics to measure acoustical and temporal aspects from audio files to train a discriminative model and improve the synthesis process of a deep neural model.

This thesis is important for the implementation of a methodology to simplify the relationship between generative models and musicology. Additionally, the implemented model presents the opportunity for the application of systematic procedures, automation of transcriptions using music notation, aural skills training for students in music and improving the implementation of deep neural networks using CPU instead of GPU due to the advantages of convolutional networks by processing audio files as vectors or matrix with a sequence of notes.

Acknowledgement

I want to take a moment to acknowledge all the support from my family. I want to thank my mother, who has shown unconditional love and support over these years. I want to thank my dad, who has supported my decisions and respected them even with disagreements.

I want to acknowledge the BSPAI research group for the constructive criticism and for allowing me to collaborate with every individual.

I want to acknowledge my advisor Winston Percybrooks, for his opinions, ideas, patience, and for correcting my errors through all these years. Thank him for being a role model, for all the support and guidance and lastly, for listening to the ludicrous idea of starting this journey one day at the hall of the laboratories in our faculty.

I want to acknowledge the lectures and lessons from all my professors during my academic formation.

Finally, I want to thank life for giving me this opportunity.

> *I learned that I don’t have to know all the answers and when I don’t, I don’t have to pretend that I do. That single lesson transformed my life. It made me willing to ask for help and it made me willing to be open to it when I need it… I allowed myself to be vulnerable.*
>
> — **Simon Sinek**

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

Artificial intelligence (AI) systems are seen as one of the most versatile tools for optimization problems [9, 21], forecasting [15, 75, 64, 70] and learning meaningful representations of data [69, 80]. AI has been used through the decades in music to understand the factors involved in the music composition process by humans and the improvement of scientific knowledge for the automation of related tasks [79, 41].

The application of AI for solving these tasks becomes suitable due to the complexity of the concept of music and its interdisciplinary relationship with other fields such as physics, acoustics, physiology, psychology, sociology and arts [24, 79, 10].

Most approaches in the literature focused on learning meaningful representations for the data using supervised learning [1] techniques. Many techniques derive from a set of rules formulated by an expert or statistics models with curated data. These sets of rules or knowledge learned based on the experience of an expert are valid from a qualitative stand and limit the scope and implementation of computational models.

However, this limitation can be avoid using massive amounts of data and great computational resources for studying the applications of AI in music, but this is not a workable solution for laboratories and university. This strategy is impractical due to the excessive cost of assets and continuous hours needed for the learning process of relevant features and the high data dimensionality of audio samples in the time domain.

In the research community, there is a tendency to show results supported by the knowledge of an expert with ambiguous relationships to the data. This thesis aims to analyze and find expressive descriptors related to the music composition process with a quantitative method using discriminant features to improve the generation of synthetic samples.

[1] Machine learning method with the goal of predict the value of an outcome measure based on several input measures along with their corresponding target values.

1.2 Motivation

According to the International Federation of the Phonographic Industry (IFPI), global streaming revenue has grown by a 42% compound annual growth rate (CAGR) since 2015, compared to the entire recording industry's 9% CAGR [62]. Previous market conditions are favorable to the opportunity for the automation of music-related tasks, such as building information retrieval tools suitable for copyright laws and international norms related to intellectual property.

The thesis contributes with a comprehensive framework[2] to improve the operational concept of music data for applications related to analysis and synthesis supported by machine learning methodologies. This framework could improve the teaching quality and learning process instructed with educational tools [35] for students in music theory, music notation and aural skills training [3] with improved peer-to-peer interaction.

Although this research may be limited to music applications, it also opens the possibility to the implementation and generation of synthetic dataset for medical disciplines where the problem of unbalanced datasets is present [47] due to the disproportion of healthy patients and patients with a confirmed diagnosis.

The thesis aims to build a model for a music system capable of analyzing and synthesizing music recordings using an expressive descriptor [4]. The system has different different preprocessing stages, data transformations and classification systems using machine learning methodologies. The development of this system will contribute to understanding meaningful representations of the data for machine learning applications in music.

1.3 Problem Statement

How to model and validate the performance music style from a composer or musician using expressive parameters extracted automatically from audio files supported by music theory?

[2] Software that is developed and used by developers to build applications.

[3] Aural training is a skill related to identifying if notes are in tune, rhythm of an interpretation is set accordingly with the metric of a score.

[4] Vector array created using signal level features extracted from the audio samples.

1.4 General System Description

The following system is proposed using a similar architecture to a generative adversarial network as shown below in Figure 1.1. The blue block is focused on the classification of the genres using a descriptor vector and the yellow block is focused on validating the descriptor vector using a synthesis model to generate new samples with similar features. The system was trained using samples from mazurka, prélude, etude, nocturne, sonata, scherzo, ballade, impromptu and miscellaneous genres.

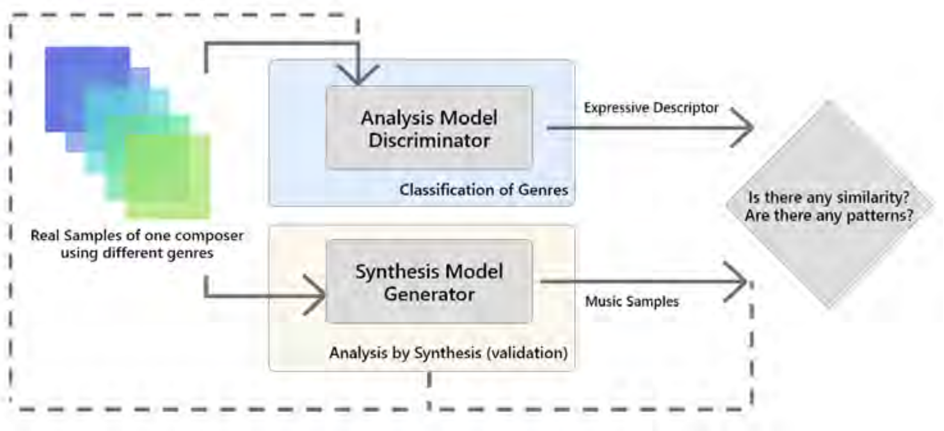

Real Samples of one composer using different genres
Analysis Model Discriminator
Classification of Genres
Expressive Descriptor
Is there any similarity? Are there any patterns?
Synthesis Model Generator
Music Samples
Analysis by Synthesis (validation)

Fig. 1.1: General system description

1.5 Objectives

1.5.1 General Objective

To build a model for a Music system capable of analyzing music recordings in order to extract expressive markers [5].

1.5.2 Specific Objectives

- To identify and extract audio features related to expressive audio information.
- To use the extracted audio features to model expressive markers and relate them to the corresponding music score.
- To test and validate the performance of the model using a pre-existing dataset, and comparing it with models from the literature.

[5] Markers in music are notations related to the dynamics used by the composer in the music score.

1.6 Scope

- The system will aim to define objective expressive markers quantitatively as opposed to subjective ones commonly used in literary works. These objective markers will then not consider the psychological aspects of a musical interpretation.

- The features extracted at the information retrieval stage are limited to the timing, dynamics and articulation. The features, are used by the implicit model.

- The use of samples with a length over 1 minute or more to capture patterns was limited and restricted. Therefore, one dataset was used for the experimentation, as opposed to the original task of analyzing five datasets, each containing music performances using piano and violin as solo instruments.

- In the state of the art, tests were performed using only classical music datasets; At least one of the datasets will be of classical music interpreted by contemporary artists.

1.7 Limitations

- Due to the nature of the music theory, validation could be necessary of counselling music experts' revision of any meta-heuristic parameter in the implicit model to validate the system.

- The system will be focused on classical music using piano and violin instruments without considering ensemble or orchestral musical performances.

- Any meta-heuristic parameter used in the implicit model does not have to be an unbiased estimator.

Theoretical Framework 2

2.1 Variable types and terminology

Vectors are denoted by lower case bold letters such as $\mathbf{x}$, and all vectors are assumed to be column vectors unless the dimensions refer to row vectors. A superscript T denotes the transpose of a matrix or vector, so that $\mathbf{x}^T$ will be a row vector. Uppercase bold letters, such as $\mathbf{M}_{(n \times o)}$, denote matrices. Dimensions of matrices are added below its instance using parenthesis to improve readability. The notation $(w_1, ..., w_M)$ denotes a row vector with M elements, while the corresponding column vector is written as $\mathbf{w} = (w_1, ..., w_M)^T$.

The notation $[a, b]$ is used to denote the closed interval from a to b, that is the interval including the values a and b themselves, while (a, b) denotes the corresponding open interval, that is the interval excluding a and b. Similarly, $[a, b)$ denotes an interval that includes a but excludes b.

The $M \times M$ identity [1] (or neutral) matrix (also known as the unit matrix) is denoted $\mathbf{I}_M$, which will be abbreviated to $\mathbf{I}$ where there is no ambiguity about its dimensionality. It has elements I_{ij} that equal 1 if $i = j$ and 0 if $i \neq j$.

A functional is denoted $f[y]$ where $y(x)$ is some function. The notation $g(x) = O(f(x))$ denotes that $|f(x)/g(x)|$ is bounded as $x \to \infty$. For instance if $g(x) = 4x^2 + 8$, then $g(x) = O(x^2)$. The expectation of a function $f(x, y)$ with respect to a random variable x is denoted by $\mathbb{E}_x[f(x, y)]$. In situations where there is no ambiguity as to which variable is being averaged over, this will be simplified by omitting the suffix, for instance $\mathbb{E}_{[x]}$. If the distribution of x is conditioned on another variable z, then the corresponding conditional expectation will be written $\mathbb{E}_x[f(x)|z]$.

Similarly, the variance is denoted $var[f(x)]$, and for vector variables the covariance is written $cov[x, y]$. Additionally, N values $\mathbf{x}_1, ..., \mathbf{x}_N$ of a D-dimensional vector $x = (x_1, ..., x_D)^T$, can be combined into a data matrix $\mathbf{X}$ in which the n^{th} row of $\mathbf{X}$ corresponds to the row vector x_n^T. Thus the n, i element of $\mathbf{X}$ corresponds to the i^{th} element of the n^{th} observation x_n.

[1] In group theory Americans called the null element the identity element.

2.1.1 Elements of Algebra

Fundamental concepts in algebra useful to describe musical structures are group, ring, field, module and vector space [6].

A group G is a nonempty set that follows the properties of associativity and commutatitivity:

$$(a+b)+c = a+(b+c)\ (associativity)$$
$$a, b \in G, a+b = b+a\ (commutativity)$$

In a multiplicative way this can be written as:

$$(a \cdot b) \cdot c = a \cdot (b \cdot c)\ (associativity)$$
$$a, b \in G, a \cdot b = b \cdot a\ (commutativity)$$

A ring R is a nonempty group that in addition to the previous definitions also follows the distributive law:

$$a \cdot (b+c) = a \cdot b + a \cdot c\ and\ (b+c) \cdot a = b \cdot a + c \cdot a\ (distributive\ law)$$

Let R be a ring such that $(R \backslash \{0\}, \cdot)$ is a group. Then R is called a division ring. if the division ring follows the commutative property, then is called a field.

A Module is defined as an associative ring within a nonempty set M with a binary operation $+ \cdot$ following these assumptions:

$$(M, +)\ is\ also\ a\ commutative\ group$$
$$For\ every\ r \in R, m \in M,\ there\ exists\ an\ element\ r \cdot m \in M$$
$$r \cdot (a+b) = r \cdot a + r \cdot b\ for\ every\ r \in R, m \in M$$
$$r \cdot (s \cdot b) = (r \cdot s) \cdot a\ for\ every\ r, s \in R, m \in M$$
$$(r+s) \cdot a = r \cdot a + s \cdot a\ for\ every\ r, s \in R, m \in M$$

These concepts and theorems are described in the literature like the presented in [6, 60, 45], but some of them are omitted or assumed to be the core of theoretical frameworks used by researchers.

2.1.2 Elements of Statistics

The most frequent used statistics in the literature like the presented in [6] are listed in Table 2.1

Name	Definition	Feature measured
Empirical distribution function	$F_n(x) = n^-1 \sum_{i=1}^{n} 1\{x_i \leq x\}$	Portion of obs. <x
Minimum	$x_{min} = min\{x_1, .., x_n\}$	Smallest value
Maximum	$x_{max} = max\{x_1, .., x_n\}$	Largest value
Range	$x_{range} = x_{max} - x_{min}$	Total spread
Sample mean	$x = n^{-1} \sum_{i=1}^{n} x_i$	Center
Sample median	$M = inf\{x : F_n(x) \geq 1/2\}$	Center
Sample quantile	$q_\alpha = inf\{x : F_n(x) \geq \alpha\}$	Border of lower 100%
Lower and upper quartile	$Q_1 = q_{1/4}, Q_2 = q_{3/4}$	Border of lower 25% upper 75%
Sample variance	$s^2 = (n-1)^{-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$	Variability
Sample standard deviation	$s = +\sqrt{s^2}$	Variability
Interquartile range	$IQR = Q_2 - Q_1$	Variability
Sample skewness	$m_3 = n^{-1} \sum_{i=1}^{n} [(x_i - \bar{x})/s]^3$	Asymmetry
Sample kurtosis	$m_4 = n^{-1} \sum_{i=1}^{n} [(x_i - \bar{x})/s]^4 - 3$	Flat / sharp peak

Tab. 2.1: Descriptive statistics

2.1.3 Data representations

The plots used to analyze, compare, and visualize might depend on the metric used for comparison among multiple audio samples or artist. Among them are line charts, bar charts, boxplot, timeseries as presented in [37].

Additionally, piano rolls are used for convenience as a quantization technique for the representation of raw music waveforms [59, 67]. The piano roll itself is presented in Figure 2.1 as a sequence of notes over time. The Y-axis represents the frequency of the note [2] over time, where the first character **C** represents a note and the second character **4** indicates the octave [3] of the respective note. The X-axis represents the times [4], where the respective pitch is being played.

[2] The pitch indicates how high or low the frequency of the note is being played.
[3] A group of twelve pitch used by the western musical system.
[4] A beat represents the relative position of the strongest accent or unaccented note.

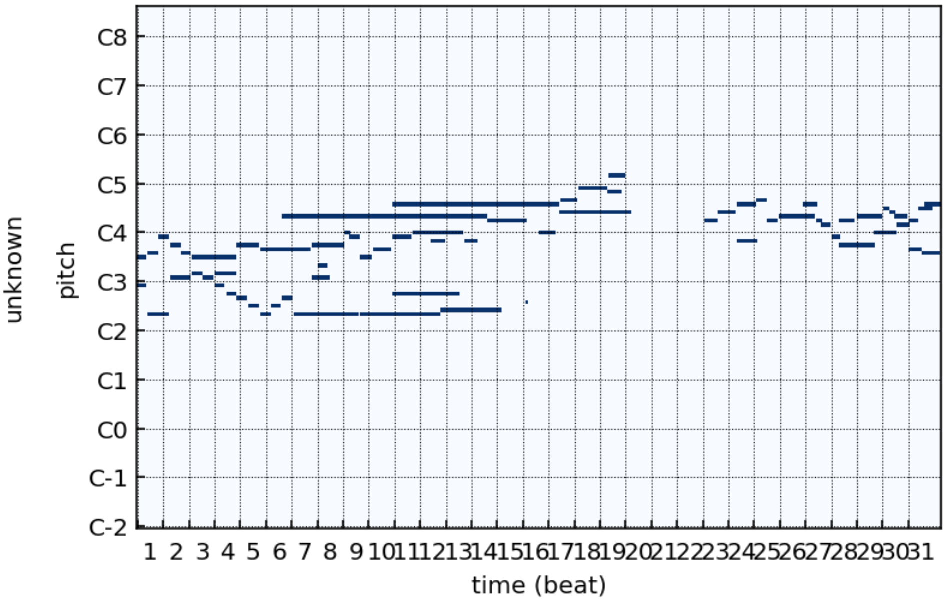


Fig. 2.1: Pianoroll representation

Since this thesis is limited to working with classical music composed using a piano or raw monophonic music waveforms, the data can be represented using music score notation as presented in Figure 2.2, this image is rendered using music21 and musescore, which are open-source toolkits used in computational musicology.

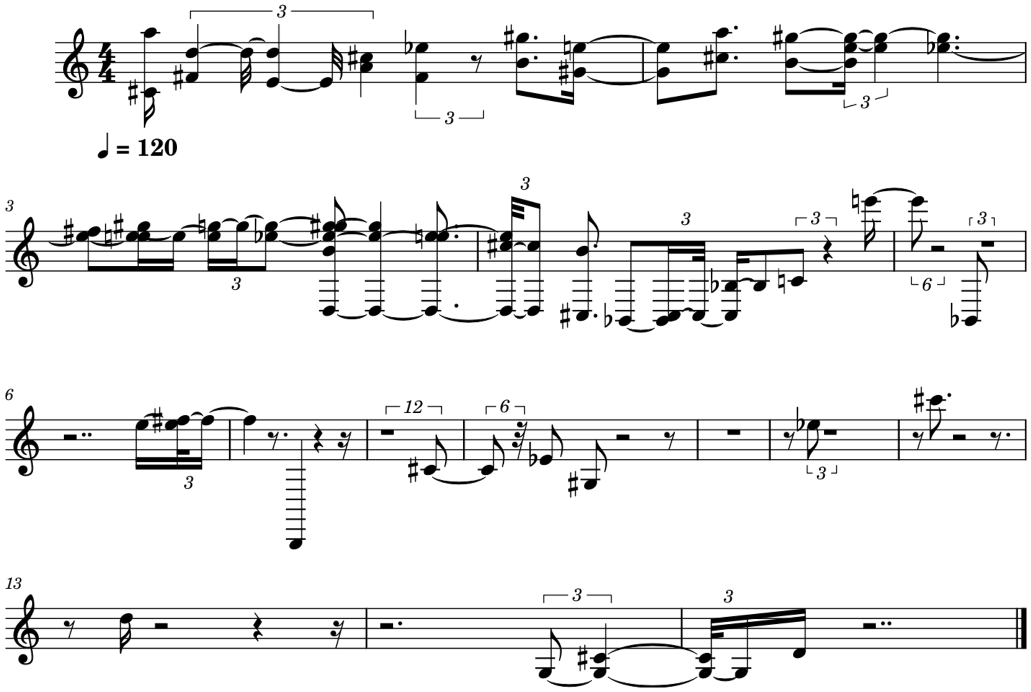


Fig. 2.2: Score notation representation

2.1.4 Expressiveness in Music

The communication of expressive content by music can be studied at three different levels: considering composer message, performer expressive intentions and listener perceptual experience [17].

In general expressiveness refers both to the means used by the performer to convey the composer message and to his own contribution to enrich the musical message. Additionally, Music performance includes all the human activity that lies between the symbolic score and the music instrument [17].

Other authors prefer the broader term expressive intentions that include emotion, affections as well as other sensorial and descriptive adjectives or actions. However, In this thesis the term expressiveness is used to model the relationship of the features with the score notation. Therefore, the term expressiveness will be associated with the concept of expressive descriptor, which refers to the signal level features extracted from music audio samples.

2.1.5 Music Performance

The concept of music performance is associated with the variations a music performer is doing when he plays. Also this concept is referred by other authors as expressive parameter.

There are three layers [17]:

- Physical Information: timing or performers movements. This information can be represented by numbers.

- Symbolic Information: the score, where the notes are represented by symbols in the common music notation.

- Expressive Information: related to the affective and emotional content of the music.

Thus, characteristic variations or patterns attributed to a certain artist is considered as individual **style** [79].

State of the Art 3

3.1 Expressive Music Performance

Expressive music performance refers to a meaningful set of features or data representations that convey the most relevant attributes of a music composition over the variations of the performance of the artist. For the analysis of these sets of features computational models have been proposed in the literature [29] [78] [77] [79]. The computational models try to describe structure or systematic deviations over the performance indicated by the notation of the score.

A robust computational model should have two parts or sub modules, one for the analysis and classification of the most relevant features and another part focused on the synthesis of these features to produce musical representations such as score notations or audio samples.

Most computational models to analyze expressiveness in music are supported by a supervised set of rules proposed by an expert in musicology. Other computational models are supported by data mining techniques using multiple recordings or compositions by one or different musicians.

3.2 Computational Models to Analyze Music Performance

There are different approaches based on supervised and unsupervised [1] learning techniques used for the implementation of computational models capable to analyze relevant features in music performances. The computational models used for the analysis are focused on learning ways to represent the data, and can be grouped in the following paradigms:

- Linear basis model (LBM).

[1] Machine learning methodology with the goal of finding relationship in data when there are not targets labels or ground truth baseline.

A linear basis modelling framework (LBM) uses a linear dependency between a set of numeric descriptors and the musical score as proposed by M. Grachten and G. Widmer [29]. Therefore, this a supervised learning approach. This model is focused on the dynamic markers of the musical score. The authors define a musical score as an ensemble of sequences as shown in the Figure 3.1 [29]. The musical score has some explicit information that can be split into simple functions like the p and f markers which are modeled using a weighted function for representing the structure of the performance.

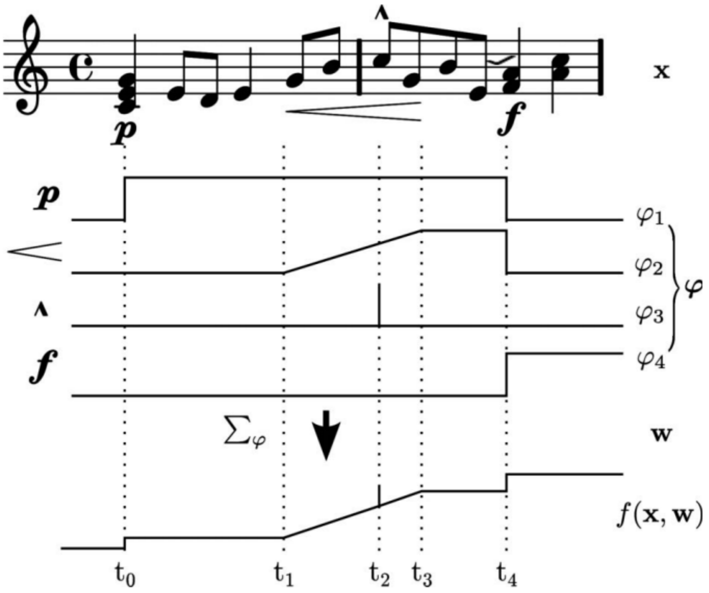


Fig. 3.1: Example of basis functions using the LBM [29].

The parameters for modeling the music performance are classified as constant, impulsive, and gradual. Depending on the timing of the annotation marker, these parameters are represented using a step function, impulse or a ramp plus a step function in the respective order [29].

As the authors suggest: "The central idea behind LBM is that it provides a way to determine the optimal influence of each parameter using a set of basis functions, in the approximation of the target" [29]. Being the target, the function that holds a relationship between dynamic markings of the musical score and the music performance.

- Nonlinear basis model (NBM).

A nonlinear basis model (NBM) uses an unsupervised learning approach to infer useful representations analyzing the data over time, in other words the model does not have a baseline to make decisions, so it needs to learn about it using the data. The nonlinear model proposed in [10] uses a multiple feed forward neural network (FFNN).

Evaluating the model can be done using a variance-based sensitivity analysis as proposed by [10] to consider the effects of each of the different basis functions for modelling the performance. This evaluation can be used to understand in a qualitative way the relationship between the dynamic markings and the performance in music [24].

- Deep learning models.

  In [28] a review about deep learning techniques for audio feature extraction is listed. Authors apply a note centric approach to have a flexible time analysis of the pitch and the relatives notes to the centered. Authors proposed to use a principal component analysis to reduce the data dimensionality and limit the usage of computational resources.

  Also in [28] feature extraction techniques are used to retrieve the meaningful information as an image, this is done to exploit the power of image representations by deep learning models in the literature.

  In [77] an algorithm is proposed to analyze the melody using a set of rules that describe and predict local timing, dynamics, and articulations related to the notes being played by the performer. The algorithm provides a relationship between the context and the set of notes. But authors use a psychological perspective in order to find a feasible and practical model, referring to it as a problem of finding partial descriptive models.

  Referring to the timing, the algorithm considered in [77] uses the inter onset interval (IOI). The length between two consecutive notes which is relative to the tempo used by the performance. To analyze the performer style, authors use the loudness level of the MIDI and articulation by the performer as a set of weighted functions formulated using the musical notes as a dictionary.

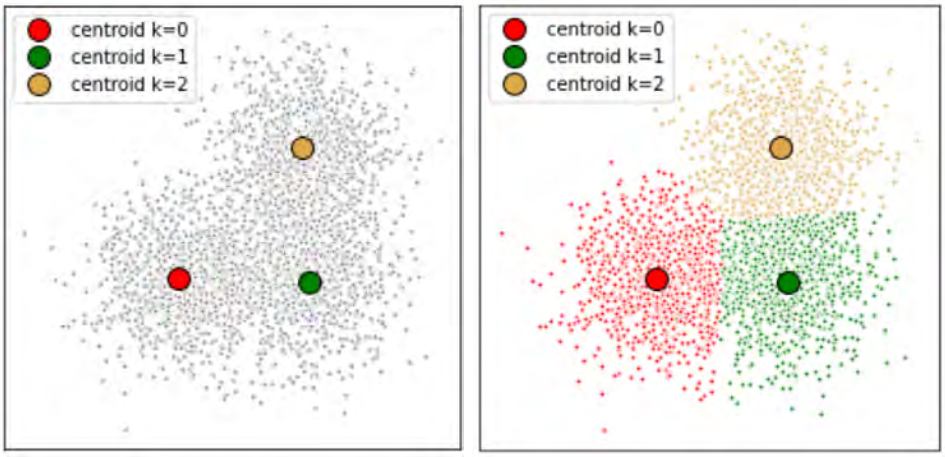
3.3 Clustering Techniques

Clustering might be a standard procedure for exploratory analysis using the music features that are more relevant, where objects in the same cluster are as similar as possible and objects in different clusters are as dissimilar as possible. The only caveat is that objects grouped in the same cluster might be different to the one's represented by the actual music label. Therefore, the clusters provide a means for generalizing over the data and their features only.

3.3.1 K means clustering

This is an unsupervised hard clustering method which assigns the n data objects $o_1, ..., o_n$ to a predefined number of exactly k clusters $C_1, ..., C_k$. Initial clusters are iteratively reorganized by assigning each data point to its closest centroid and recalculating the cluster centroids until there are no further changes [2], which means a similarity measure is need for comparing the data points [12, 8]. The optimizing criteria in the clustering process is the sum of squared error E between the objects in the clusters and their respective centroids $cen_1, ..., cen_k$ as follows:

$$E = \sum_{i=1}^{k} \sum_{o \in C_i} d(o, cen_i)^2 \tag{3.1}$$



(a) Setting centroids in space

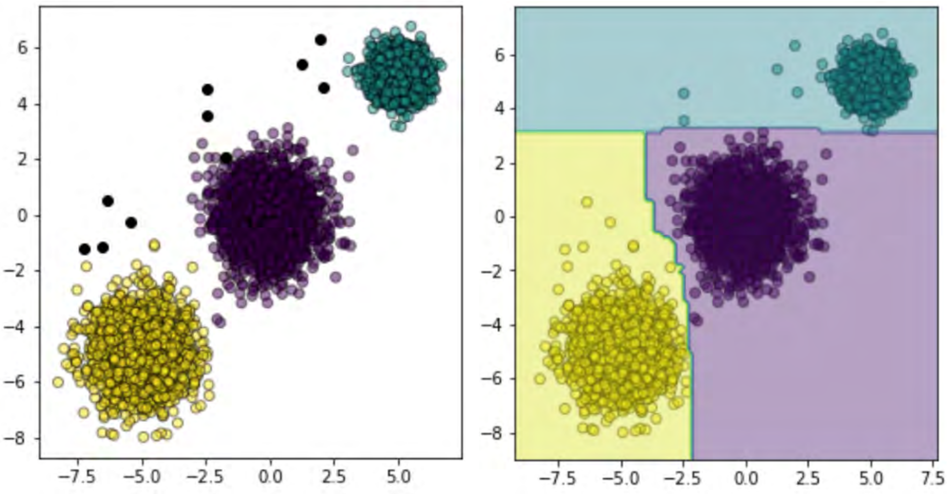
(b) Clustering data points

Fig. 3.2: K means clustering

3.3.2 Agglomerative clustering

This is a hierarchical clustering technique that creates a dendrogram, which is a tree structure containing k clusters or sets of partitions between 1 and n, where n is the number of data points or features to cluster [46, 16]. Thus, each point or feature begins as its own cluster and progressively join two closest clusters using a threshold distance to reduce the number of clusters by 1 until $k = 1$ [3].

[2] Charts are generated using Scikit-learn [61] library and a random generator with Numpy library.
[3] Charts are generated using Scikit-learn [61] library and a random generator with Numpy library.



(a) Adding new instance to clusters (b) Clustering new points

Fig. 3.3: Agglomerative clustering

3.4 Classification Methodologies

Expressive descriptors describe meaningful relationships for the musical data from a genre or an artist, but once the selection of these features is completed, it is a task for classification algorithms to test their utility for real world applications.

3.4.1 Linear discriminant analysis (LDA)

The LDA is a dimensionality reduction technique that transform a series of features into a lower dimensional space by maximizing the ratio of separability or distance between different classes using the mean of a class regarding the other classes and minimizing the variance of a class [71].

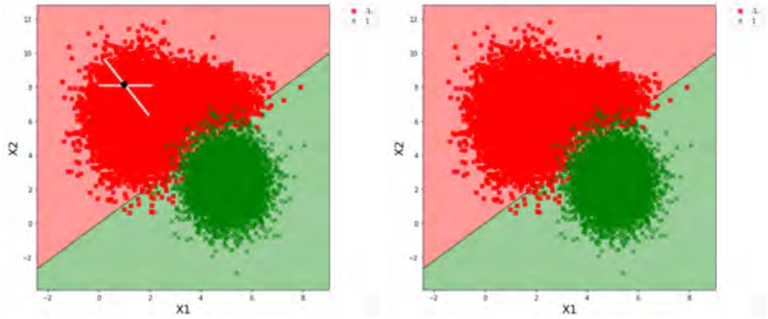
Previous description can be summarized in general if the original data represented as a matrix ($\mathbf{X} \in R^{N\times M}$) is reduced by projecting it onto a lower dimensional space of LDA ($V_k \in R^{M\times k}$) using the following equation:

$$Y = XV_k \tag{3.2}$$

3.4.2 K-nearest neighbor (KNN)

Nearest neighbor classifiers use a set of observation in the training set T closest in input space to x samples to form $\hat{Y}$, which is the prediction or the expected result for a classification. The algorithm uses a similarity metric to measure the closer members, which are defined as K groups [27]. After the selection of these K groups, the algorithms start a majority voting among all the k points within a group until a validation criteria is reached like the example presented in Figure [4] 3.4 [42] . $\hat{Y}$ is defined as follows:

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i \tag{3.3}$$

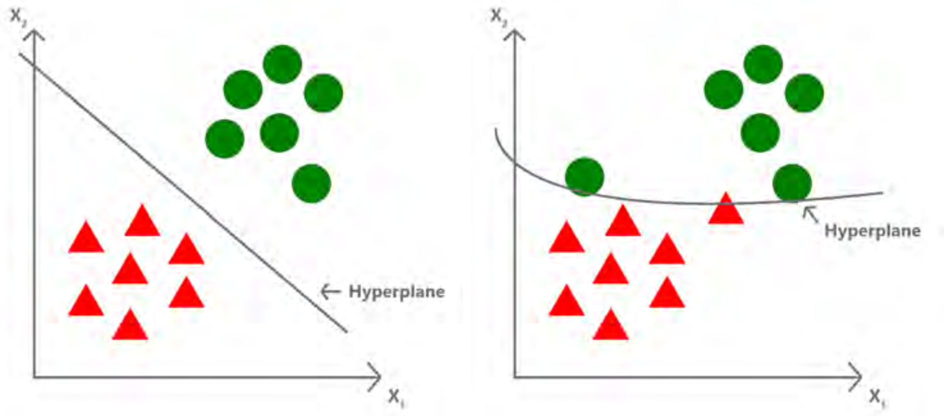


(a) New point added for testing (b) point assigned to class

Fig. 3.4: K-nearest neighbor classifier

3.4.3 Support vector machines (SVM)

Support Vector Machine is a machine learning technique used for classification and regression. Support Vector Machine is based on supervised learning, which classifies points to one of two disjoint half-spaces [34]. It uses nonlinear mapping to convert the original data into higher dimension space [18]. Its objective is to construct a function, which will correctly predict the classes or groups to which the new point belongs. Using an appropriate nonlinear mapping, two data sets can always be divided by an hyperplane. Hyperplanes separate the tuples of one class from another and define a decision boundary [2].



(a) Linear separable classes case (b) Nonlinear separable classes case

Fig. 3.5: Support vector machines

[4] Charts are generated using Scikit-learn [61] library and a random generator with Numpy library.

3.4.4 Decision Tree (DT)

This technique uses a set of rectangles to partition the feature space and then fit a simple model with the most frequent labels in it. Given a training set S with data attributes set $A = \{a_1, a_2, ..., a_n\}$ and a nominal target attribute y from an unknown fixed distribution D over the labeled instance space. A decision tree DT follows the goal to classify labels with a minimal error or misclassification rate over the distribution D [40, 66].

$$\epsilon(DT(S), D) = \sum_{<x,y>\in U} D(x, y) \cdot L(y, DT(S)(x)) \tag{3.4}$$

where $L(y, DT(S)(x))$ is the zero one loss function defined as:

$$L(y, DT(S)(x)) = \begin{cases} 0 \; if \; y = DT(S)(x) \\ 1 \; if \; y \neq DT(S)(x) \end{cases} \tag{3.5}$$

A decision tree example for the classification of the iris dataset is available in [31] and the visual representation can be found in Scikit-learn [61] documentation and is shown in figure 3.6.

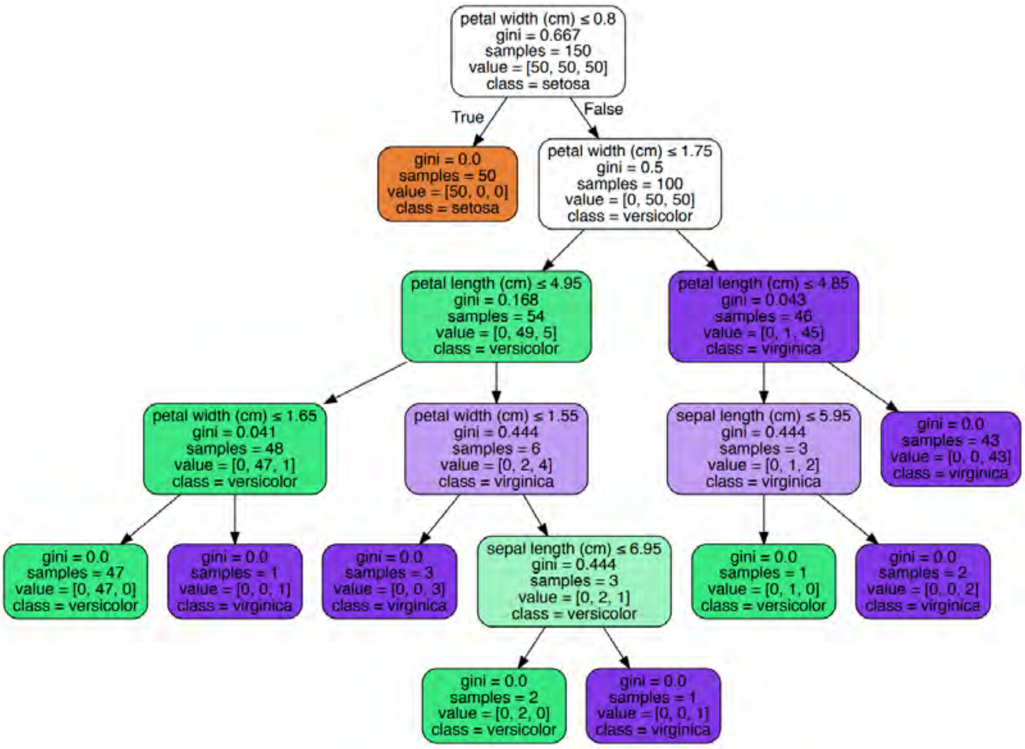


Fig. 3.6: Decision tree using iris dataset

3.4.5 Artificial Neural Networks (ANN)

Artificial neural networks tend to mimic the way humans learn by heart. Otherwise by learning from examples, which means that in order to make predictions or classifications, the ANN should be trained using examples. One of the most popular implementation is the feed-forward neural network, also known as the multilayer perceptron [7].

Neural networks use basis functions that are based on linear combinations of fixed nonlinear basis functions, so that each basis function is itself a nonlinear function of a linear combination of the inputs, where the coefficients in the linear combination are adaptive parameters. Because they can be adjusted during training.

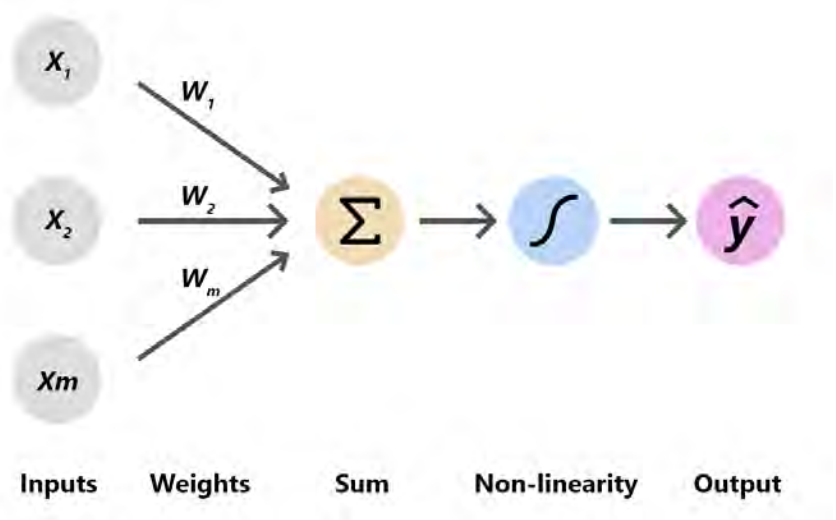


Fig. 3.7: The Perceptron

Previous statement can be observed in the equation 3.6 where $\hat{y}$ is the output, and g is the non linearity or activation function. the other part of the equation represents the linear combination of inputs.

$$\hat{Y}(x) = g(\sum_{i=1}^{m} x_i w_i) \tag{3.6}$$

This model is used in the literature for information retrieval tasks, onset detection [43] or genre classification as the application in [23] using a database of popular songs listened in Colombia with similar acoustical features within the Merengue, Salsa and Vallenato genres.

3.4.6 Convolutional Neural Networks (CNN)

Convolutional Neural Networks take advantage that the input consists of images and they reduce the architecture size in comparison to an ANN [44]. The layers of a CNN have neurons arranged in 3 dimensions: width, height, and depth (refers to the color channel in an image). The neurons in a layer will only be connected to a small region of the layer, instead of all of the neurons in a fully-connected manner. The output of a CNN is a single vector representing the classes or categories predicted or classified.

Convolutional neural networks are invariant to certain transformation performed over the data (images) used as input, making them really useful for classifying or predicting new data or observations outside of the ones used in the training process. CNN uses three mechanisms: local receptive fields, weight sharing, and subsampling [7]. In the convolutional layer the units are organized into planes, each of which is called a feature map.

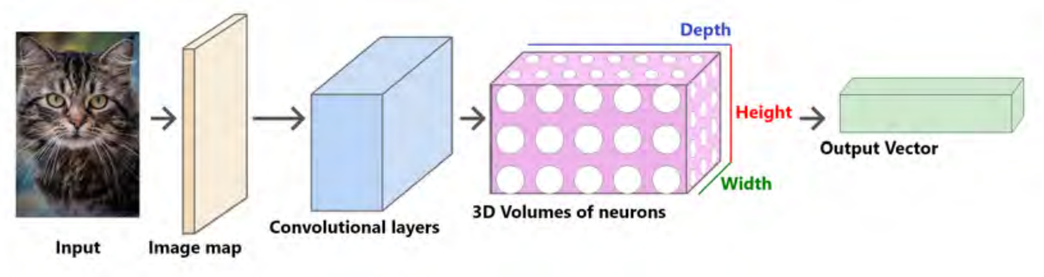


Fig. 3.8: Convolutional network representation

Units in a feature map take inputs only from a small subregion of the image, and all of the units in a feature map are constrained to share the same weight values. Thus each unit becomes a feature detection map for the same input at different locations. By applying shifting over the inputs and feature maps the CNN learns how to deal with translations and distortions.

This model can be used for prediction [11] and the classification of genres such as traditional western, Latin music and African music collections [14].

3.5 Modeling with Expressiveness

Understanding the concept of expressiveness in music and music performance should be clear up to this point, but some might think of the word "emotion" to explain that a piece composed with a slow tempo might sound sad. The variation from the typical performance constitutes expressiveness, regardless of its emotion. Different strategies were present in this chapter for building computational models and techniques to find structural parameters using clustering techniques or classification methodologies.

Models are employed to evidence and abstract some relations that can be hypothesized, discarding details that are felt to be not relevant to what is being observed and described. Models can be used to predict the behavior in a certain condition and compare these results with observations. In this sense, they serve to generalize the findings and have both a descriptive and predictive value [17].

Most significantly, linear basis models, limited by the knowledge of an expert to formulate a set of functions to model the structure of the performance and nonlinear basis models infer useful representations over time but lack a meaningful way to showcase the relationship of the model build with the expressiveness of the performance. Additional resources defined a set of techniques supported by machine learning methodologies as deep learning models. This model might have similar pitfalls as the nonlinear models, but one advantage over the nonlinear models comes from exploiting the power of image representations.

Method	**Utility**
LBM	Rule based approach supported by an expert knowledge
NBM	Inference of relationships between score and features
Kmeans clustering Agglomerative clustering	Inference of relationships among data to find similarities
LDA	Classification of a set of data points or features
KNN	Classification and requires a similarity metric
SVM	Classification and requires large data representations
DT ANN CNN	Classification of a set of data points or features

Tab. 3.1: Literature summary

In summary, the limitations of the computational models in the state of the art showcase an opportunity to build and develop computational models supported by machine learning methodologies as long as the relationship with the expressive descriptors is clear using systematic validations and studying the relation between performance and operational variables by using a system capable of analyzing and synthesizing music recordings.

4 Research Methodology

4.1 General Description

The proposed system shown in the Figure 4.1 is organized using different blocks. The first block is the unit for the Data Processing and the Data Augmentation. This unit is built on top of pyAudio [25] and Librosa [48]. This unit has the task of performing the feature extraction and selection of useful data representations that could potentially capture useful information over the music samples.

The next block is the Analysis Model, which has the task of testing the usefulness of the previous features by classifying them into groups. This block uses different algorithms like K-means and agglomerative clustering, which can be grouped as linear models. In addition to those algorithms, the Analysis Model uses SVM, DT, and ANN, which can be grouped as Non-linear models.

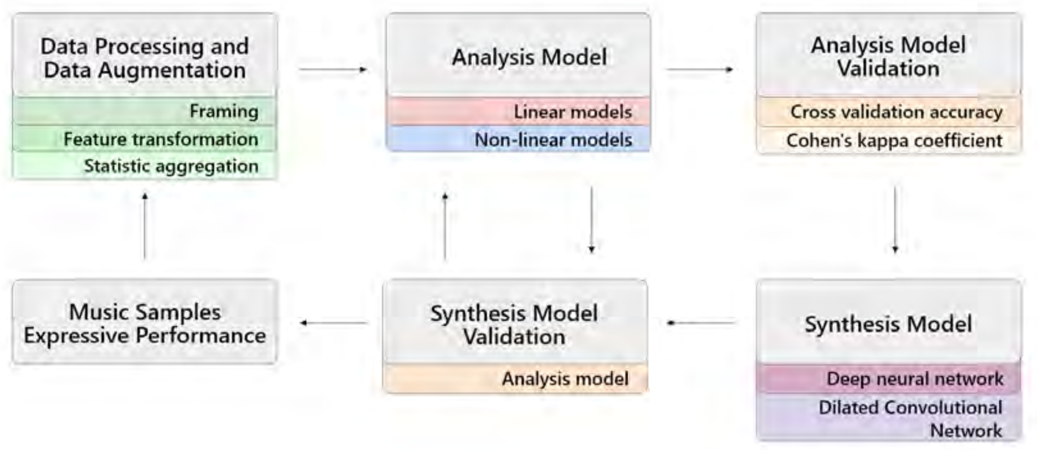

Data Processing and Data Augmentation
Framing
Feature transformation
Statistic aggregation
Analysis Model
Linear models
Non-linear models
Analysis Model Validation
Cross validation accuracy
Cohen's kappa coefficient
Music Samples Expressive Performance
Synthesis Model Validation
Analysis model
Synthesis Model
Deep neural network
Dilated Convolutional Network

Fig. 4.1: General diagram of the system implemented for the analysis and synthesis of music samples

The next unit is the Analysis Model Validation, which task is to validate the results coming from the Analysis Model. The block unit uses cross validation accuracy and the Cohen's kappa coefficient. The silhouette coefficient [72] also was implemented, but its usage only makes sense when using unlabeled data. The silhouette is used to select the optimal number of cluster based on a metric of dissimilarity among the groups of features.

Last, but not least is the Synthesis Model, which uses the raw data and the features transformations with the representative information of the music to generate new music with a similar style. The synthesis model uses a Deep Neural Network built with Dilated Convolutional networks for the learning process. Other modules are subproducts or subtasks of the previous modules described above.

4.2 Databases

4.2.1 Data Acquisition

For the data acquisition stage, several resources were taken into consideration like the databases from the International Symposium on Methodologies for Intelligent Systems (ISMIS) and the International Conference on Music Information Retrieval (ISMIR), but since the length of multiple music datafiles are short, the decision to collect the data from an streaming service was taken. The streaming service that allows downloading the raw music data is Apple Music using a paid subscription.

The databases have been used by other researchers for the task of music correspondences with score notation and piece identification, which means that they have been through a curation and labeling process by an expert. The databases selected are mainly focused on classical music.

Database 1

Considering the different variations and different genres within classical music, the Magaloff corpus was selected. This dataset comprises performances of virtually the complete Chopin piano works, as played by the Russian-Georgian pianist Nikita Magaloff (1912-1992). The music was performed in a series of concerts in Vienna, Austria, in 1989, on a $B\ddot{o}sendorfer\ SE$ computer-controlled grand piano [51] that recorded the performances onto a computer hard disk. The recorded data contains highly precise measurements of the times any keys and pedals were pressed and released, and the intensity with which they were pressed. The data set consists of 9 genres, 158 pieces, adding up to over 320.000 performed notes, almost 10 hours of music.

The Magaloff corpus is an unbalanced genre collection dataset of 9 genres shown in Figure 4.2. formed with songs all being encoded in mp3 format. The frequency and bitrate of these files are 44.100 Hz and 16 bps respectively. The Magaloff dataset has a minimum metadata file with the year of publication and genre associated to the audio sample file.

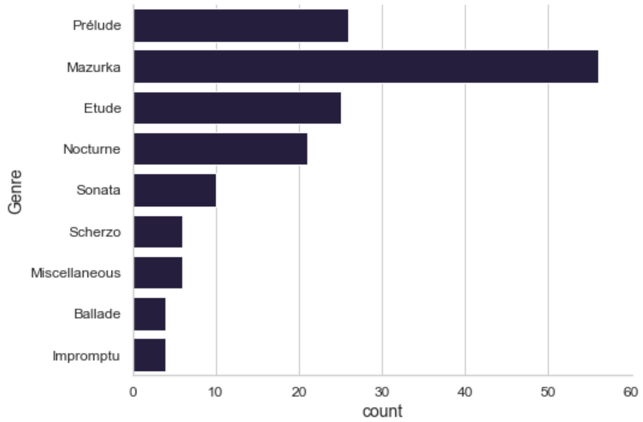


Fig. 4.2: Count of audio sample per genre using the Magaloff Dataset

Database 2

This database takes into consideration modal variations and it had been used for musical score sheet notation and music alignment. The dataset can be found as Multimodal Sheet Music Dataset [19], which comprises 479 precisely annotated solo piano pieces by 53 composers, for a total of 1,129 pages of music and about 15 hours of aligned audio. Additionally, the music records were synthesized by doing renditions from these scores. The database also uses nearly the complete solo piano works by Frederic Chopin and commercial recordings by famous concert pianists.

Database 3

This database is composed of about 200 hours of piano performances captured with fine alignment ($\approx$ 3 ms) between note labels and audio waveforms. It is hosted by TensorFlow in partnered with organizers of the International Piano-e-Competition. The dataset can be found as the MAESTRO [32] (MIDI and Audio Edited for Synchronous Tracks and Organization).

Database 4

This is a collection of 330 freely-licensed classical music recordings called MusicNet licensed under public domain with over 1 million annotated labels indicating the precise time of each note in every recording, instrument, and the note position in the metrical structure of the composition. The labels are acquired from musical scores aligned to recordings by using dynamic time warping. The labels are verified by trained musicians with a labeling error rate of 4% estimated. The MusicNet labels are offered to the machine learning and music communities as a resource for training models and a common benchmark for comparing results.

Database 5

This database consists of classical piano midi files hosted on the kaggle website, which is known for different public and private competitions, the dataset contains compositions of 19 famous composers and a total of 295 files.

4.3 Signal Processing and Data Augmentation

4.3.1 Feature Extraction

The frequency and bitrate of music files often are 44.100 Hz and 16 bps respectively to save the quality and replicate it seamlessly over different devices. Since the system should have an objective methodology to analyze the music performance, some transformations to extract signal level features were tested. This decision was made to check the best transformations for working on classical music and due to the limitations of modeling music files as a time sequences, which represents a highly computational cost. Therefore, the system should have a way to represent the data in an efficient and meaningful manner.

4.3.1.1 Zero crossing rate

The zero-crossing rate as shown in equation 4.1 indicates the frequency of signal amplitude sign change [5] where the k denotes the discrete time index. The time interval between successive time indices is defined by the inverse sampling frequency $1/fs$. The signal is segmented into frames $x[,k]$ of length k. The signum function $sgn()$ yields 1 for positive arguments and 0 for negative arguments. The zero-crossing rate [54] is a rough measure of the noisiness and the high frequency content of the signal.

$$t_{zcr}[\lambda] = \frac{1}{2(K-1)} = \sum_{k=1}^{K-1} \left| sgn(x[\lambda, k]) - sgn(x[\lambda, k-1]) \right| \tag{4.1}$$

4.3.1.2 Spectral contrast

The spectral contrast describes the ratio between the magnitudes of the peaks and the valleys within sub-bands of the frequency spectrum [3, 50] as shown in Equation 4.2.

$$t_{sc}[\lambda] = log(\frac{\frac{1}{\alpha N}\sum_{i=1}^{\alpha N} x_{k,i}}{\frac{1}{\alpha N}\sum_{i=1}^{\alpha N x_{k,N-i+1}}}) \tag{4.2}$$

where N is the total number in $K-th$ sub-band, $k[1,6]$ this correspond to a frequency domain division into six Octave-scale sub-bands. The strong spectral peaks roughly correspond with harmonic components, while non-harmonic components, or noises often appear at spectral valleys [38].

4.3.1.3 Spectral centroid

The spectral centroid determines the frequency bin around which the highest amount of spectral energy is concentrated [54]. It is defined as the center of gravity of the magnitude spectrum as shown in equation 4.3.

$$t_{cent}[\lambda] = \frac{\sum_{\mu=0}^{M/2} \mu|X[\lambda,\mu]|}{\sum_{\mu=0}^{M/2} |X[\lambda,\mu]|} \tag{4.3}$$

Where the μ denotes the discrete frequency index. Additionally, for symmetry reasons, the summations range from 0 to M/2 only, this is a constraint for the Nyquist theorem. Lower values correspond to dull sounds, whereas higher values denote brighter sounds.

4.3.1.4 Tonal centroid features

It is a planar representation of pitch relations modelled by small distances on the plane [30]. The information is processed as a finite matrix representation with a twelve different pitch classes. The six dimensional tonal centroid vector, ζ_n, for time frame n is given by the multiplication of the chroma vector, c, and a transformation matrix Φ as shown in equation 4.4.

$$\zeta_n(d) = \frac{1}{||C_n||}\sum_{l=0}^{11} \Phi(d,l)c_n(l) \qquad \substack{0 \leq d \leq 5 \\ 0 \leq l \leq 11} \tag{4.4}$$

where l is the chroma vector pitch class index and d denotes which of the six dimensions of ζ_n is being evaluated. The transformation matrix Φ represents the six dimensional space. This transformation can be useful for chord recognition.

4.3.2 Feature Selection

The first step of the feature selection procedure is finding audio features containing discriminating information. The second step consists in finding an optimal sample window for the training and synthesis model; for that reason the criteria selected was finding the base note of each individual sequence for any audio sample. The base note is used as reference for getting the optimal length of samples to learn structural patterns in the music composition process.

4.3.2.1 Time Dimension

Although data sampling does not consider any high-level knowledge about music structure, in the literature, often an interval of 30s [74, 49] is analyzed. This number might be associated to legal issues, because in some countries audio excerpts of 30s length could be freely distributed. Three short-term windowing (framing) were tested to find an optimal length for working with classical music files.

Non-overlapping frames were sampled with an offset of 2 seconds to limit the presence of silence or amplitudes close to zero. The first sample window was set to 5 seconds, but it was not feasible to identify the base note. The second sample window was set to 9 seconds, it was feasible identifying the base note. The third sample window was 30 seconds, but performing Fourier analysis requires a high computational cost in comparison to the previous sample window.

For each individual feature, the distribution of the data is described. Additionally, the data distribution of the raw features and the normalized features are compared to observe the influence of the scaling process over the unsupervised learning approach. The previous comparison is done to analyze the changes over the audio features. Additionally, the previous step has to be performed for the validation purpose of the modelling process using machine learning.

4.3.2.2 Feature Dimension

Thus, for the purpose of dimensional reduction, saving storage requirements, computing costs and increase the class separability of the genres in the datasets; the transformations described in 4.3.1 were applied over the 30s frame sequences. This process also provides a way to working with audio samples with different lengths by using as much segments of frames as possible. At the same time using the selected transformations over a sequence of frames guarantees using interpretable music features, even after the dimensional reduction.

Finally, the estimation of statistics was performed to increase the classification quality of the analysis model [52, 39]. As a result of testing different transformations and statistics aggregation [55]; these combination of parameters (feature transformations and statistics) were selected in this methodology. These are the ones that result in better accuracy for the analysis models using audio samples from classical music. The complete workflow is described in section 4.3 and is shown in figure 4.3.

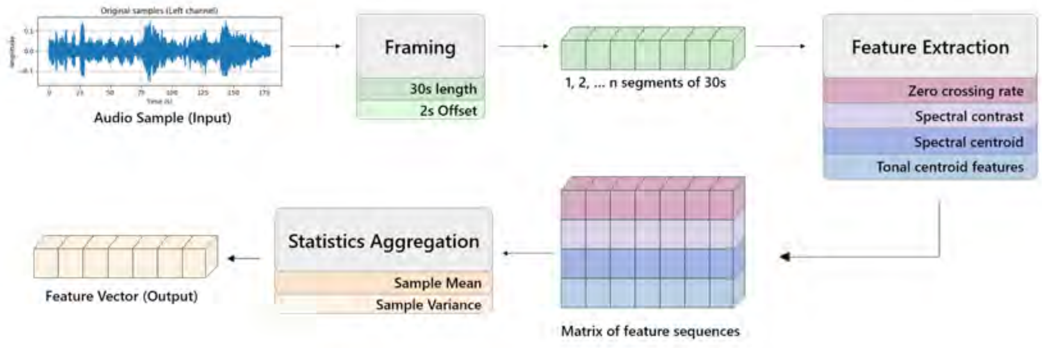


Fig. 4.3: Data processing workflow

4.4 Data Analysis over Music Parameters

The implementation of different models was tested consequently to assert the premise that machine learning algorithms or computational models can learn meaningful representations and data structural patterns. Using the previous array of features from the music files using the dataset described in section 4.2.1, some machine learning models were trained and tested most significantly to validate the utility of the suggested features as representative to contain meaningful content for classification tasks.

The metric used for the evaluation of the models was the Cohen's kappa coefficient [13], which expresses the level of agreement between two annotators on a classification problem. The metric is defined as:

$$k = \frac{(p_o - p_e)}{(1 - p_e)} \tag{4.5}$$

where p_o is the empirical probability of agreement on the label assigned to any sample (the observed agreement ratio), and p_e is the expected agreement when both annotators assign labels randomly. p_e is estimated using a per-annotator empirical prior over the class labels [4].

4.4.1 Linear Models for Classification

Clustering was performed using K-means and hierarchical agglomerative, both algorithms were validated considering different partitions (k[2,10]). The APP metric described in equation 4.7 was used instead of F-measure as external evaluation metric for measuring the effectiveness of the algorithms [12]. This is a simple version based on [76] considering $C = \{C_1, ..., C_k\}$ as the correct objects in a group; $A = \{A_1, ..., A_k\}$ as alternative objects in a group to analyze the effect of the alternatives over the correct objects in a cluster as shown in the equation 4.6 was implemented:

$$APP(C_i) = \frac{C}{A+1} \tag{4.6}$$

Which has the following constraints:

- If APP $<$ 1, thus the proportion of A objects is bigger than C in the cluster. by a 0.5 proportion.
- If APP $=$ 1, thus the proportion of A is 0.5 the proportion of C in the cluster.
- If APP $>$ 1, thus the proportion of A objects is lesser than C in the cluster.

In order to consider the effect of the k groups over the objects a scaling process is applied over the previous equation alternative as shown in 4.7. This metric is similar to the precision, but without minimizing the effects of the correct objects in a group.

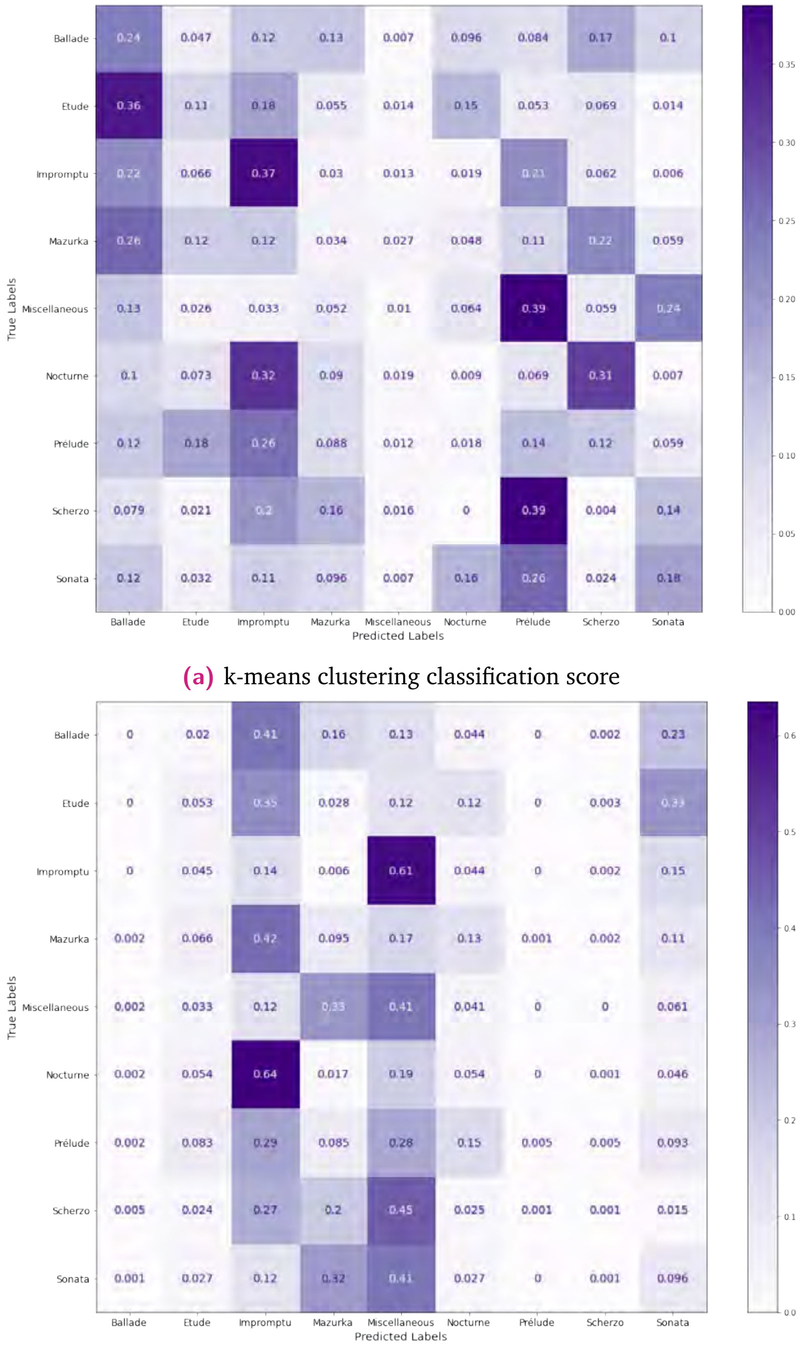
$$APP(C_i, k) = \sum_{i}^{k} \frac{APP(C_i)}{k} \tag{4.7}$$

However, the results presented in this section are limited to consider the metadata of the different genres as the baseline for the clustering process. Thus, results for the linear models are presented in figure 4.4. The accuracy and the Cohen's kappa coefficient were used to evaluate the classification of the models as presented in table 4.1. By comparison the agglomerative clustering algorithm is overestimating the genres with more audio samples in the training process and the k-means is compensating this effect by identifying correctly in testing more genres.

Algorithm	Avg. Accuracy	Cohen's Kappa [1]
K-means clustering	0.212	0.124
Agglomerative clustering	0.127	0.018
Decision trees	0.702	0.649
Support vector machine	0.642	0.580
Artificial neural network	0.703	0.646

Tab. 4.1: Analysis models classification score

[1] A kappa greater than 0.5 means that the models are capable to perform a classification better than a random classifier.



(a) k-means clustering classification score

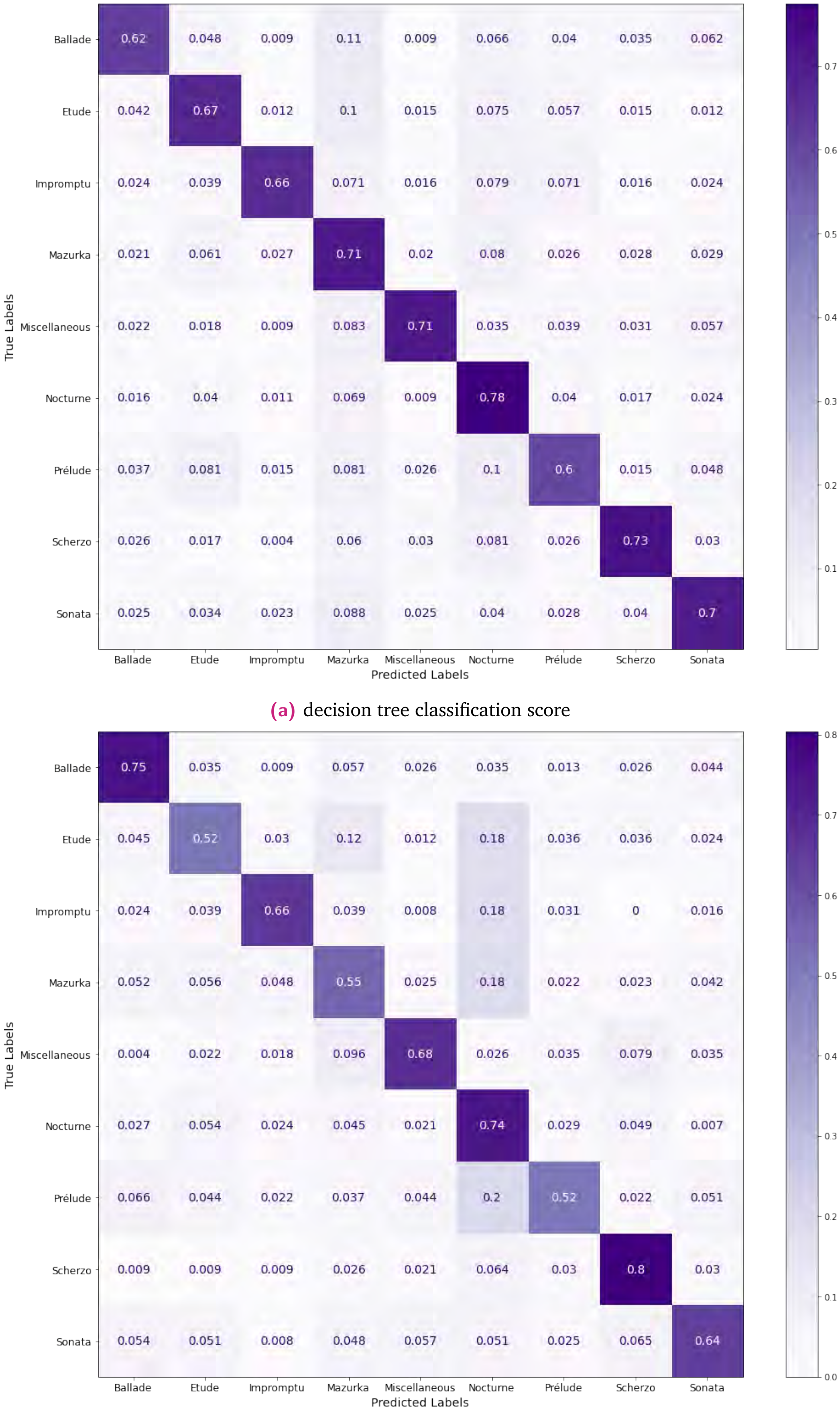
(b) Agglomerative clustering classification score

Fig. 4.4: Linear models for classification

4.4.2 Non Linear Models for Classification

The models selected for this subsection were the decision trees, support vector machines and artificial neural networks as described in subsection 3.4. Results in table 4.1 suggests that non linear models can use the selected features accordingly to the Cohen's kappa coefficient [2]. As a consequent, the analysis models might be used for the validation of the results of the synthesis model.

[2] A kappa greater than 0.5 means that the models are capable to perform a classification better than a random classifier.



(a) decision tree classification score

(b) support vector machine classification score

Fig. 4.5: Non-linear models for classification

Additionally, the features that follow the Independence and Identity Distribution (IID) assumptions are shown in table 4.2 for the validation purpose of modelling data using machine learning approaches.

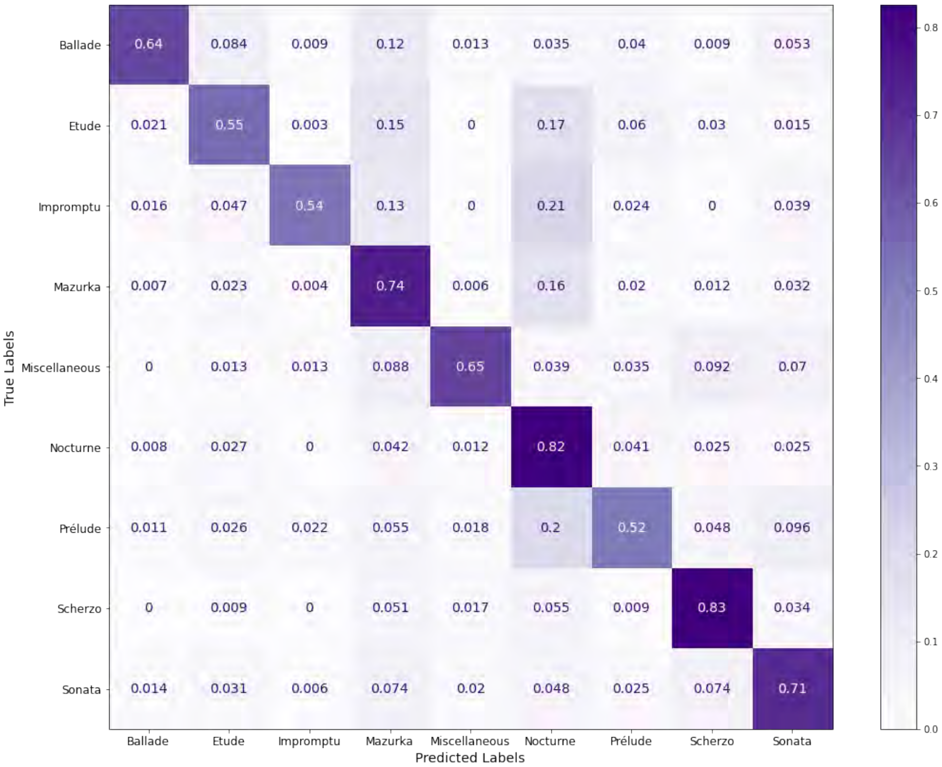
Feature	Independence	Identity	Normalized
Zero crossing rate	True	True	True
Spectral contrast	True	True	True
Spectral centroid	True	True	True
Tonal centroid features	True	False	True

Tab. 4.2: Independece and Idendentity test

For the implementation of the decision tree the Gini impurity was used as the information criteria for measuring the quality of a split. The minimum number of samples required to be at a leaf node was set to 1 and the minimum number of samples required to split an internal node was set to 2. The artificial neural network architecture is shown in table 4.3. These hyperparameters are set based on a exhaustive search considering 150 as the maximum number of iterations and a tolerance of 0.001.

Layer	Shape	Note
Dense Layer	1024	Relu activation
Dropout	1024	0.3%
Dense Layer	64	Relu activation
Dropout	64	0.1%

Tab. 4.3: Artificial neural network architecture



(a) Artificial neural network classification score

Fig. 4.6: Non-linear models for classification

4.5 Validation

Regarding the implementation of the previous models for the analysis, results summarized in table 4.1 are calculated using cross validation accuracy and the Cohen's kappa coefficient. Results for the training are within a similar range with a variation of 5% for the linear models and 1% for the non-linear models, therefore, results for the training process of each model has been omitted. The feature vector obtained in the data processing workflow (figure 4.3) is used for training the analysis model using a split [3] considering the 20% for testing.

The cross validation [63] is implemented to deal with the issues of getting favorable classifications by randomly selecting a particular pair of training and validation sets. When using cross validation a test should be held out for a final evaluation (classification), but the validation set is no longer needed when doing cross validation. A basic implementation is shown in figure 4.7 called k-fold cross validation, where the training set is split into k smaller sets. The resulting model is validated on the remaining part of the data (shown in yellow folds) computing a performance metric such as accuracy. The performance measure reported by k-fold cross-validation is then the average of the values computed in each of the splitting (k-1 Training).

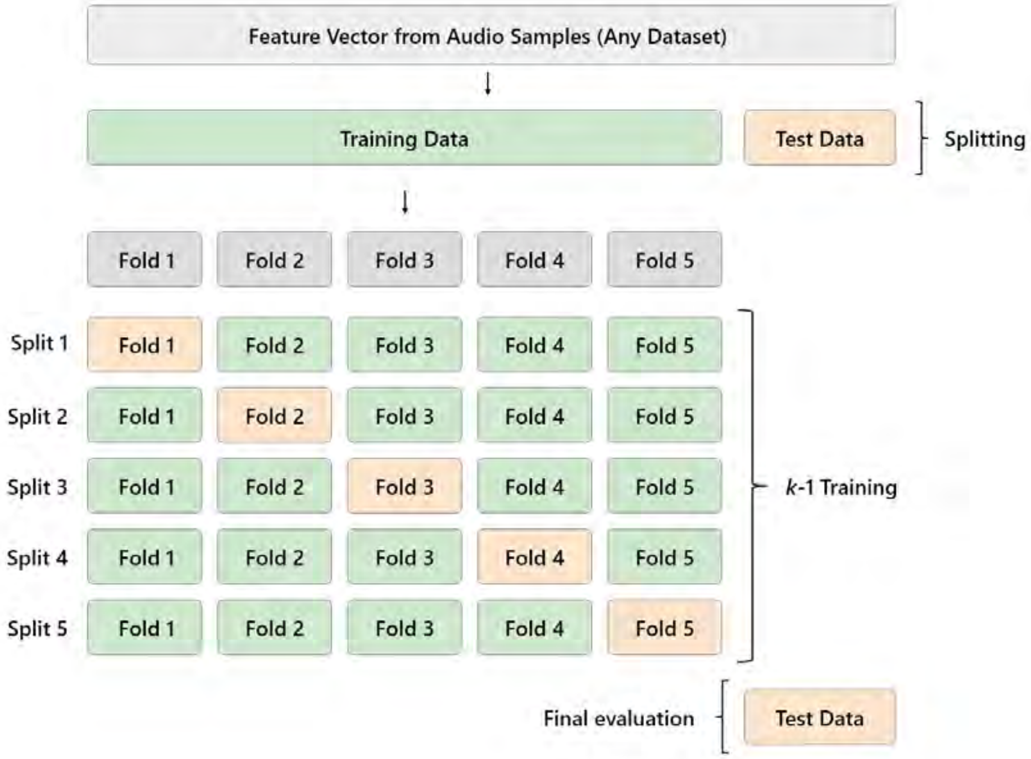


Fig. 4.7: Cross validation process

[3] The 80% of the data is used for training.

4.5.1 Data Synthesis

The synthesis model is based on the proposed in [56] named WaveNet, an audio generative model which subsequently is based on the PixelCNN architecture [57, 58]. This model learns the conditional probability distribution of the raw audio waveforms using convolutional neural networks, where the prediction of the model at timestamp t cannot depend of any of the future timestamp. For the previous reason this convolutional networks are called causal convolutions. The stack of a causal convolution is shown in Figure 4.8.

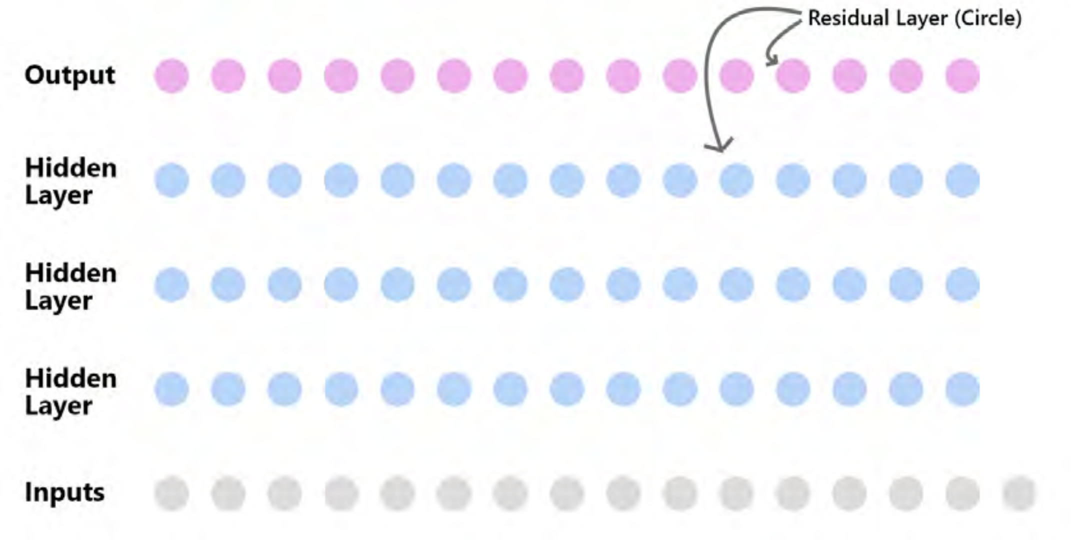


Fig. 4.8: Causal Convolution stack

The model implements residual layers [4] [5] [33] and skip connections to increase the convergence speed and propagate signals more directly to the network. The activation unit is described as:

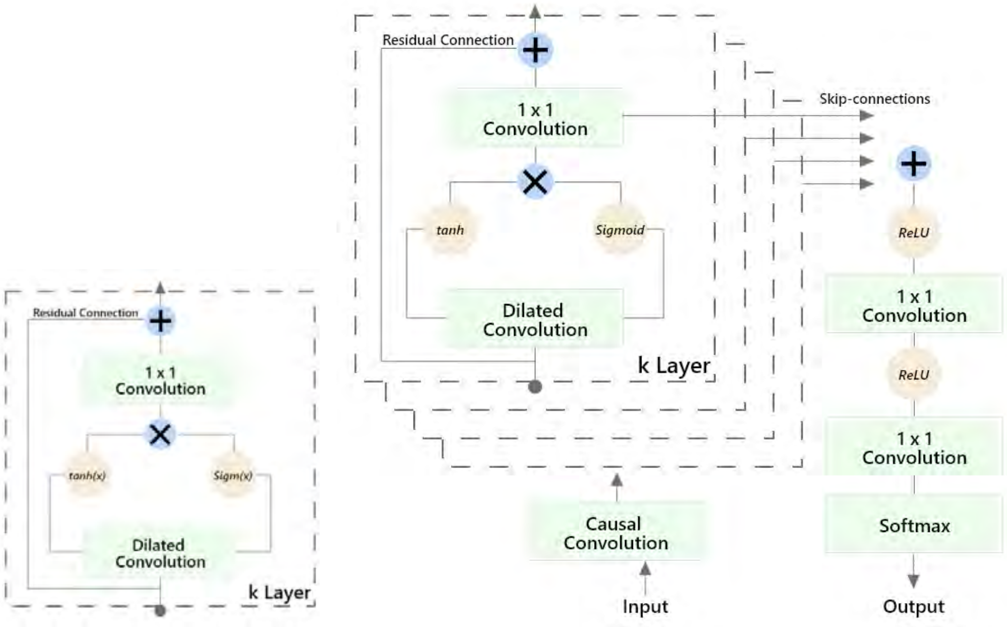
$$z = tanh(W_{f,k} * x) \odot \sigma(W_{g,k} * x) \tag{4.8}$$

where $*$ denotes a convolution operator, $\odot$ denotes an element-wise mutiplication operator, σ is the sigmoid function, k is the layer index, f and g are the filter and gate, respectively. W is the learnable convolution filter.

Considering the implementation of the PixelCNN used for the synthesis of images, the equivalent of a causal convolution is a masked convolution [57] which can be implemented by doing an element wise multiplication over the mask tensor with

[4] residuals layers are used to address the degradation problem due to the vanishing gradients in the learning process of deep network models.

[5] residual blocks are represented by dots in the causal convolution stack.



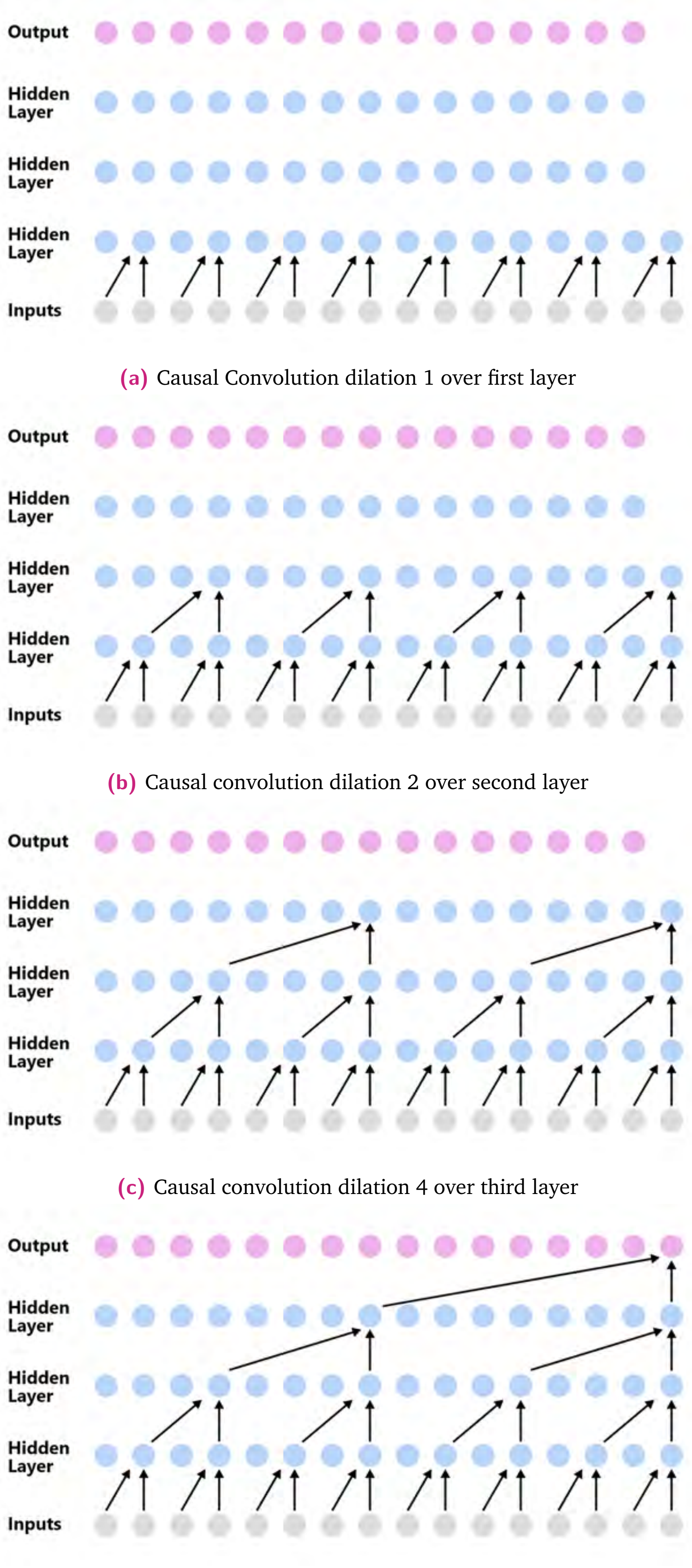
(a) Residual block

(b) Residual layer

Fig. 4.9: Residual block architecture used in WaveNet [56]

the convolution kernel. A similar implementation for 1-D data or sequences is accomplished by shifting the output of a normal convolution by a few timesteps. The process of the causal convolution is shown in Figure 4.10.

Consequently, operations like shifting or applying casual convolution over a matrix representation of a time series like a pianoroll could be implemented using machines with a general purpose CPU. Therefore, the performance for training a model with these architecture would be similar to using a machine with an specific GPU.



(a) Causal Convolution dilation 1 over first layer

(b) Causal convolution dilation 2 over second layer

(c) Causal convolution dilation 4 over third layer

(d) Causal convolution dilation 8 output

Fig. 4.10: Causal convolution process

Because models with causal convolutions do not have recurrent connections, they are typically faster to train than recurrent neural networks (RNNs) also used to model long term relationships on sequences.

A dilated convolution effectively allows the network to operate on a coarser [6] scale than with a normal convolution. This is similar to pooling or strided convolutions, but here the output has the same size as the input. As a special case, dilated convolution with dilation 1 yields the standard convolution [56].

Raw audio is typically stored as a sequence of 16-bit integer values (one per timestep), a softmax layer would need to output 65,536 probabilities per timestep to model all possible values. To make the task more tractable, authors first apply a μ-law companding transformation [36] to the data, and then quantize it to 256 possible values using equation 4.9:

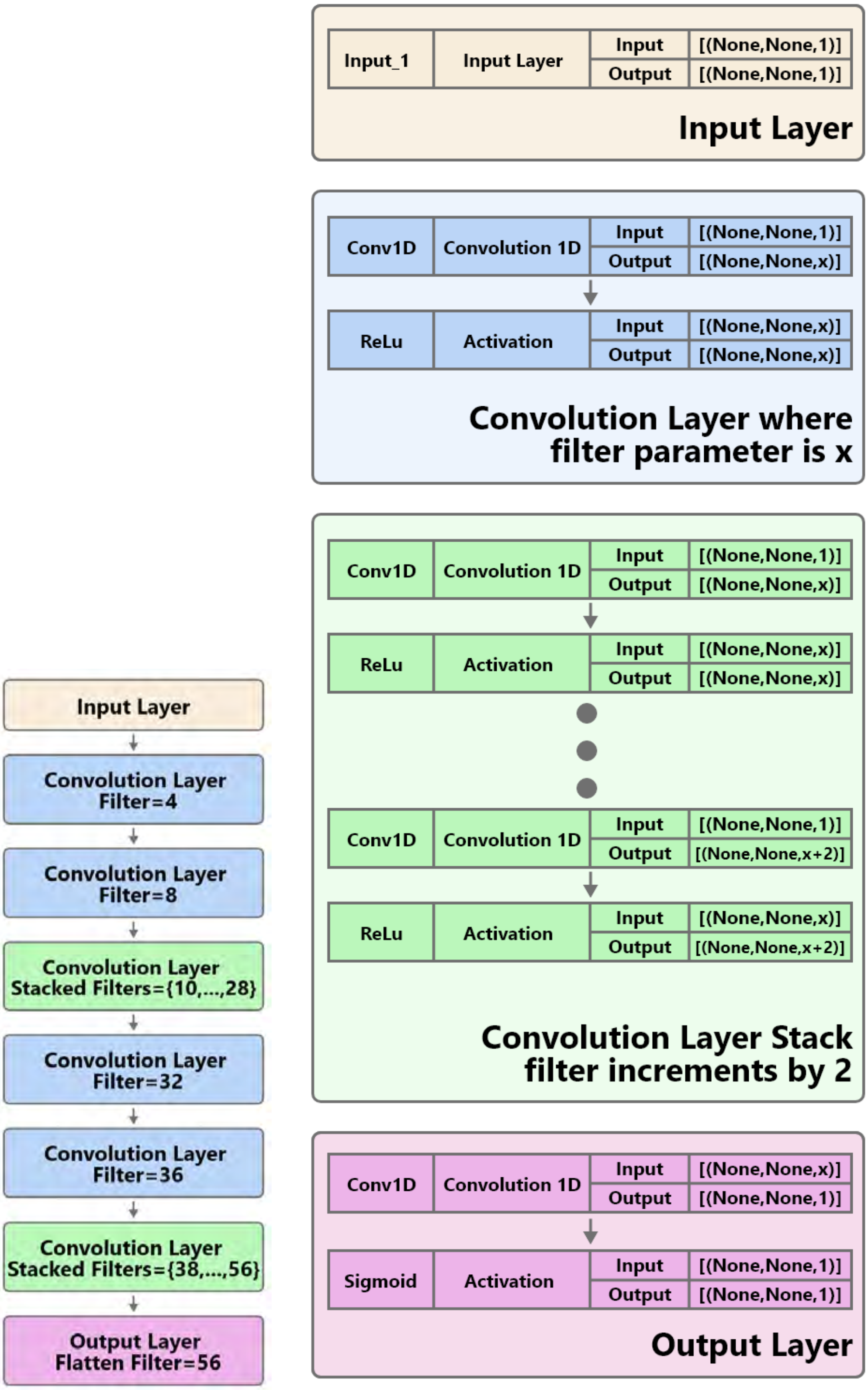
$$f(x_t) = sign(x_t)\frac{\ln(1+\mu|x_t|)}{\ln(1+\mu)} \tag{4.9}$$

The conditional probability of the samples is modelled using softmax activations functions. the softmax function generates an embedding representing the probability associated to a note. The preference for this function over others is because a categorical distribution is more flexible and can easily model arbitrary distributions while making no assumptions about their shape [56].

The synthesis model architecture implemented is shown in Figure 4.11a. Since this is a deep neural network some of the building blocks have been summarized to fit in one page, but the internal architecture of these blocks is described in Figure 4.11b. The term filter is used to describe the dimensionality of the output space of the convolution. Additionally, the model architecture for the convolutional layers is restricted to kernel sizes equals to 2, strides equals to 1, paddings set to $valid$ as implemented using keras framework and dilatation rates are equal to 2 by the power of the layer index.

A similar diagram can be generate using TensorFlow [1] and Graphviz [22], but the first one does not compress some of the layer architectures while keeping the metadata visible. For example, the stacked convolutions with increments by 2 and other parameters of the architecture making it difficult to fit in one page.

[6] Coarser refers to work using a smaller sets of samples or sub-samples in the context of the training process.



(a) model summary

(b) model architecture

Fig. 4.11: Synthesis model

The model architecture as described in the chapter did not work as expected. In fact, the generated audio files were not as fluid as the rendered editions or even audible sounds sometimes. Therefore, the addition of other preprocessing and post processing blocks were implemented.

The pre-processing stage to transform an audio file with extension type wav to a midi is performed using Melodia toolkit [67]. The post-processing is performed using the timidity toolkit [73].

Additionally the model implemented in this thesis, is powered by the music21 [7] toolkit library [53] to parse the synthetic samples using midi or wav audio files extension to score notation as shown in figure 4.13.

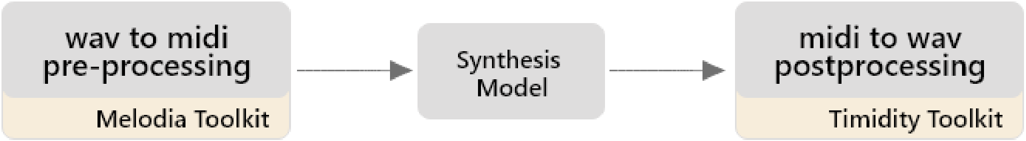


Fig. 4.12: Pre-processing and Post-processing Additional Blocks

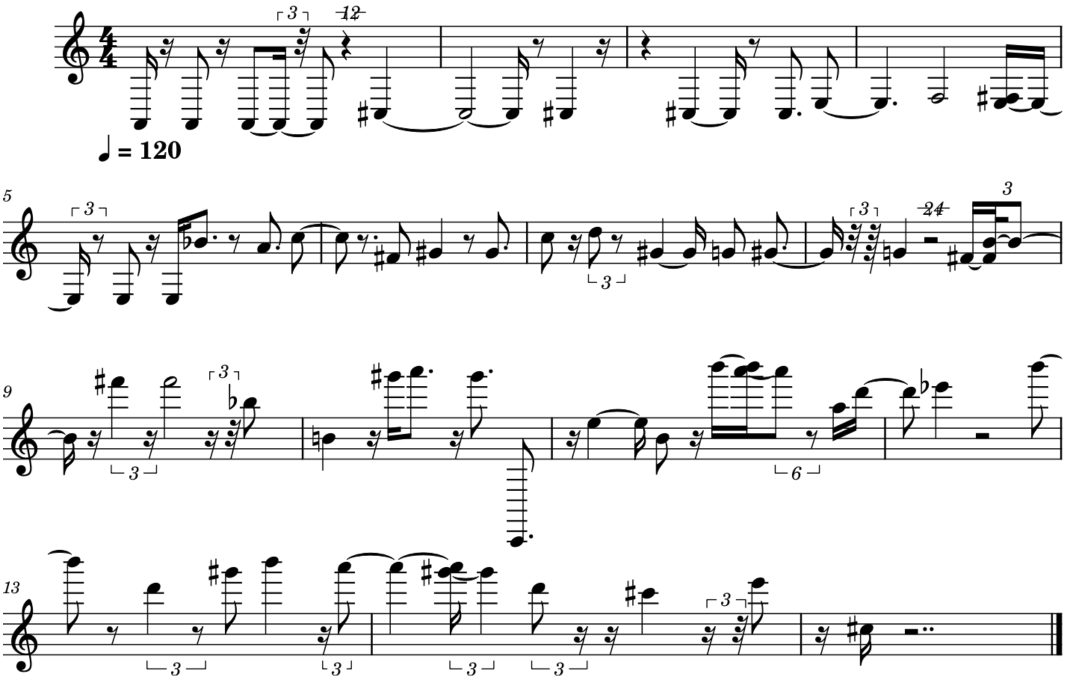


Fig. 4.13: Score Notation Synthetic Sample

[7] library developed by the MIT for computer aided musicology.

4.5.2 Design and Analysis of Experiments

Regarding the synthetic samples, the following experimentation as shown in table 4.4 was performed to test if the most relevant attributes (section 4.3.1) are present;

$$\begin{array}{llll}
RG_k & O_C & X_M & O_M \\
RG_{k+1} & O_C & X_M & O_M \\
\vdots & \vdots & \vdots & \vdots \\
RG_{k+n} & O_C & X_M & O_M
\end{array}$$

Tab. 4.4: Design of Experiments

Where the R (random assignment), G (sample groups), O (measurement), X (experimental condition) follow the symbolic convention used in [65], where k represents the synthetic sample generated for a genre. n represents the total number of genres. O_C represents the classification labeled assigned to synthetic sample using the classifier from the analysis model. X_M represents a transformation of the audio waveform to a midi file to measure in a modality the presence of expressiveness. Finally, O_M represents the measurement of some variables (pitch, step and duration of the notes) for external validation.

Experimentation workflow:

- Select one of the datasets available.
- Generate 5 audio samples for each group (genre) in the dataset.
- Generate a Feature Vector (Signal Transformation) for each sample.

Experiment Conditions:

- Seeds for the synthesis are randomly assigned.
- Model architecture remains static.
- Synthesis process is performed using a pre-trained model using the original data for each group (genre).
- Training for each group (genre) has been executed 5 times using random seeds as start sequence.
    - Framing is performed using a slice window of 30s and offset set to 2s.

For this experimentation the dataset 1 defined in section 4.2 was selected because it was the only one with audio files with longer length between 1 minutes and 5 minutes. The decision tree model was selected for the test considering the Cohen's kappa coefficient results in table 4.1. Results of the classification using the analysis model are shown in table 4.5. At first glance most of the features were assigned incorrectly.

However, the more predominant genre with right labels was the $Sonata$, this might suggest a correlation to the seed assigned randomly during the synthesis. The previous hypothesis is based on the fact that during the training process the presence of samples belonging to the $Sonata$ genre was lower in comparison to other samples, discarding the idea of overfitting during the training.

	right	wrong	samples (Feature Vector)
Ballade	7	89	96
Etude	18	77	95
Impromptu	0	96	96
Mazurka	2	93	95
Miscellaneous	10	86	96
Nocturne	2	94	96
Prelude	16	79	95
Scherzo	12	84	96
Sonata	35	61	96
Average	11.33	84.33	95.67

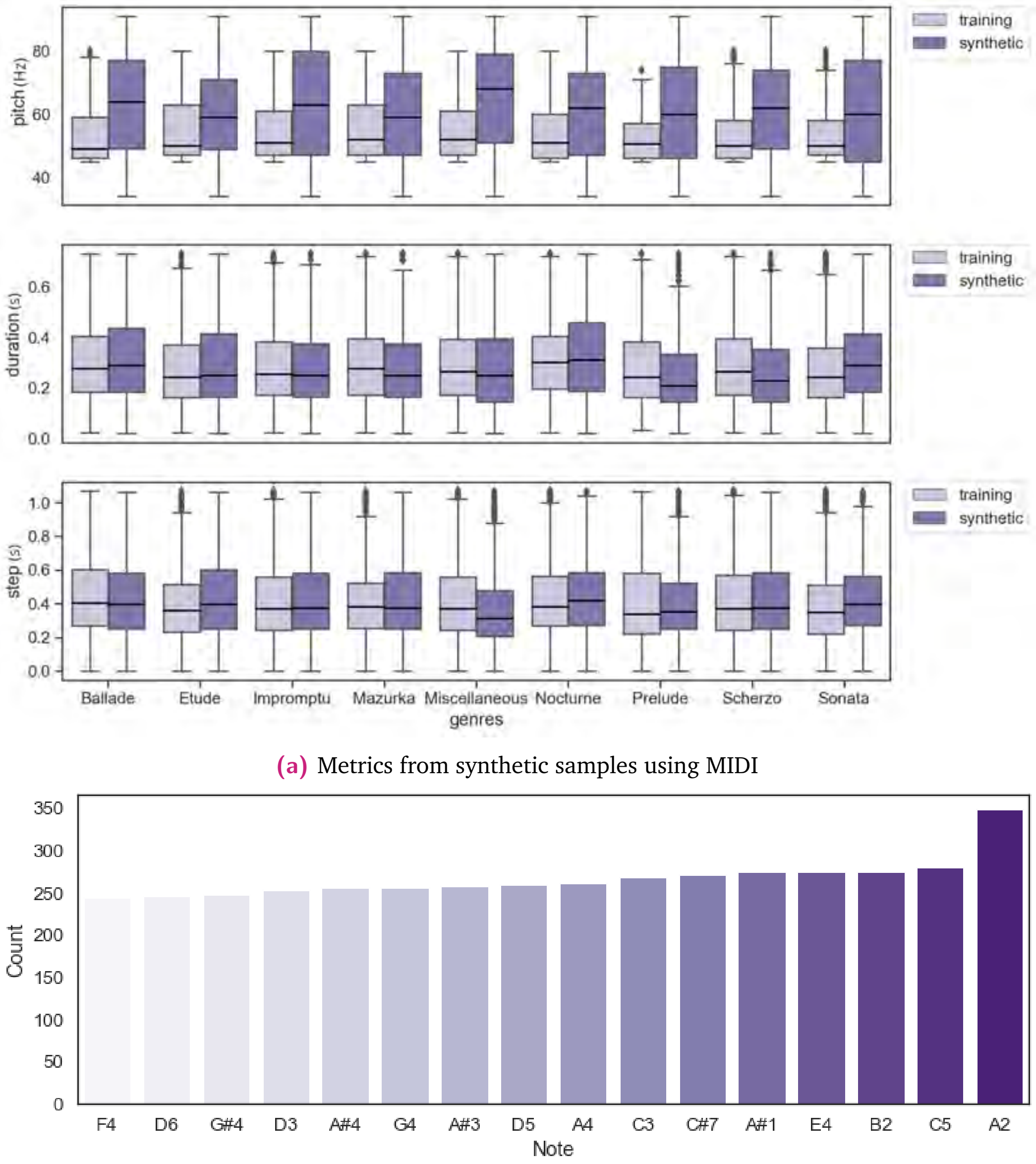
Tab. 4.5: Classification of Synthetic Samples using Wavenet

Results for the external validation metrics in figure 4.14 show an overlap among the medians [8] for each of the metrics (pitch, steps and duration of the notes) after removing the outliers using the interquartile range to filter the data. This filtering was performed setting the lower quartile to 0.25 and the upper quartile to 0.75 equation.

The $pitch$ is the perceptual quality of the sound as a MIDI note number. The $step$ is the time elapsed from the previous note or start of the track. The $duration$ is how long the note will be playing in seconds and is the difference between the note end and note start times.

[8] The F-test was also performed, comparing the variance of the pith range across different groups (genres), resulting in a p-value of 0.00; implying a strong rejection of the null hypothesis of no differences in the pitch range distribution across the different synthetic samples.

Additionally, it is noticeable the presence of silence (the absence of notes) in many instances of time. On the other hand, the most frequent notes are presented in figure 4.14b, this notes were parsed from the pitch distribution in figure 4.14a. For instance, this information might display a relationship similar to a scale, that is not the case. Actually, it just displays a range of notes used in the synthesis process.



(a) Metrics from synthetic samples using MIDI

(b) Top 15 Most Frequent Notes

Fig. 4.14: External Validation Metrics using Wavenet Model

4.5.3 Comparing Generative Models

In addition to the external validation metrics, other models were tested using the same experimentation workflow summarized in table 4.4 to provide a practical and operational context about the synthesis process and the advantage of implementing the generative model proposed in this thesis.

Results using a Recurrent Neural Network (RNN) for the synthesis trained using the Maestro Dataset [32] and the classification using the analysis model (DT [9]) are shown in table 4.6. Results show a lower rate of accuracy classification considering the proportion of the labels assigned wrongly. However, there is a difference between the wavenet model against the RNN regarding the length of the synthetic samples after the transformation to a Feature Vector length, implying that synthetic audio samples generated by the RNN models are shorter than the one's generated by the wavenet model.

	right	wrong	samples (Feature Vector)
Ballade	8	5	13
Etude	0	14	14
Impromptu	0	14	14
Mazurka	1	26	27
Miscellaneous	0	53	53
Nocturne	0	26	26
Prelude	2	30	32
Scherzo	0	18	18
Sonata	0	9	9
Average	1.22	21.67	22.88

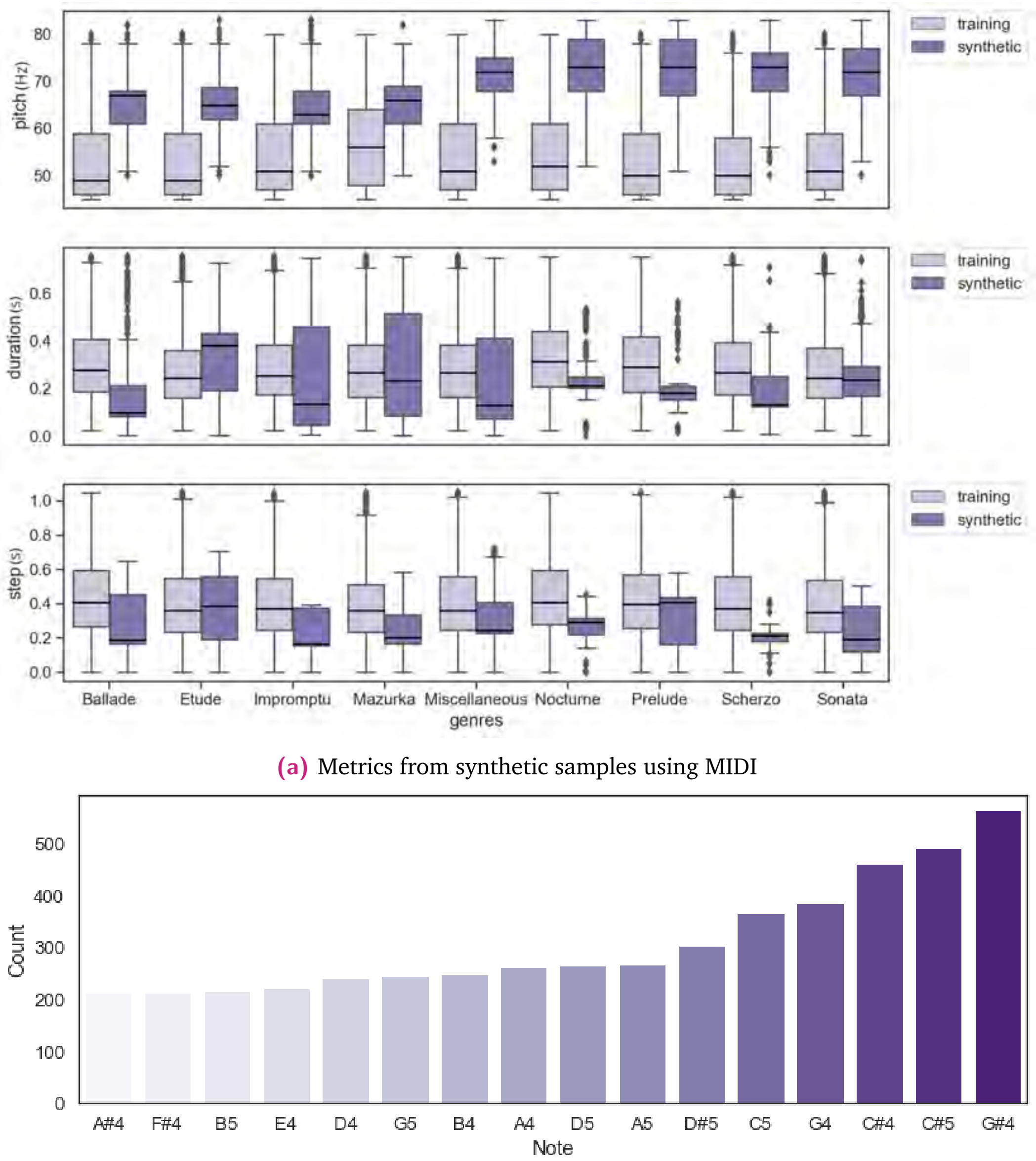
Tab. 4.6: Classification of Synthetic Samples using RNN

Simultaneously, both models are not capable to synthetize audio files with the structural patterns of $Impromptu$ genre. This might be related to the lower count of samples in the dataset as described in figure 4.2.

Furthermore, analyzing the distribution of the range of pitch over the synthetic samples generated by the RNN model, it seems to be an overlap between two groups of ranges, but, performing the F-test results in a p-value of 0.00; implying a strong rejection of the null hypothesis of no differences in the pitch range distribution across the different synthetic samples.

[9] Decision Tree model.

Lastly, after comparing the distribution of steps values by the two synthesis models as shown in figure 4.15a, there seems to be a reduction between the magnitude of steps for the notes using the RNN model. In contrast, the wavenet model has higher values for steps, this could be translated to slower compositions, which are more related to some extend of the performance in genres composed by Chopin.



(a) Metrics from synthetic samples using MIDI

(b) Top 15 Most Frequent Notes

Fig. 4.15: External Validation Metrics using RNN model

4.5.4 Evaluation Strategy for the System

It is important to emphasize the reasons behind the usage of the synthesis model and analysis model. The approach for evaluating the features and validating the quality of the synthetic samples is called Analysis by synthesis [17, 26], which seeks to explain the synthesis inference using the features of the data. Similar implementations of the method are explained in [68, 20].

The method considers the following stages and the implementation of those in this thesis:

- Synthesis of performances with systematic variations: accomplished by using different seeds in the synthesis model for each genre.
- Judgment of synthesized versions: accomplished by using the analysis model and the vector of features described in section 4.3.1 to check the relationship of the synthetic sample with the base genre.
- Control of the reliability of the judgments followed by classifications of the listeners: Accomplished by generating 5 music files per genre and asserting the prediction for the next note in a sequence only if the note is part of the main corpus of the dataset.
- Study of the relation between performance and experimental variables: Accomplished to some extent by using external validation metrics.
- Repetition of the previous steps until results converge.

For simplicity and considering the scope of the thesis, the only systematic variation is introduced by changing the seed for each synthetic audio generated. Therefore, previous evaluation tested the hypothesis regarding the utility of the proposed metrics in 4.3.1 as expressive descriptors.

5 Results Summary

1. Identification and extraction of audio features related to expressiveness using audio files.

   Features related to the expressiveness were extracted using an algorithm built on top of pyAudio [25] and Librosa [48] open source libraries for music information retrieval (MIR) to measure timing and acoustical information as described in section 4.3.1. To test or identify the usefulness of these features, they were used as input data to train different machine learning algorithms for the classification of the features setting a proportion of 20% for validation in each sample population.

   Each of the extracted features were transform or reduced to a vector array containing the mean and the variance over the normalized metrics from each sample population. This preprocessing stage allows the classifiers to reach an accuraccy between 64% to 70% and a cohen kappa score over 0.6. This tests were performed under difficult conditions by using an unbalanced dataset as pointed in chapter 4.

2. Modeling Audio files with expressive markers and showing the relationship using the musical score notation.

   Deep Learning methodologies were used to implement and develop a model capable to generate synthetic sample with expressiveness in music using signal level features as described in 4.5.1 in conjunction with dilated causal convolutional layers with the architecture shown in Figure 4.11a. the details for each block are shown in 4.11b to guarantee the reproducibility of this work.

   The model is probabilistic and autoregressive. The audio files represent time series, that subsequently are parse to a note matrix as shown in subsection 2.1.3 with the time duration or absence of the performed notes, this is a preprocessing stage to quantize the data. This is a limitation found over the experimentation because the synthesis of samples using the raw audio was not possible accordingly to preliminary experimentation and results.

The model can synthesize audio files using the expressiveness in music audio files by creating a discriminant model that is going to be trained using these features as discriminant features, which subsequently are used to estimate the probability associated with the next notes considering a previous sequence of notes.

Results suggest smoother sequence of notes in the synthetic audio files and notes within a range of pitch as shown in section 4.5.2. Accordingly to the p-values of 0.00 using the F-test to compare the variance over the synthetic samples across the different groups of study of musical genres.

Additionally, the model can generate audio files using the MIDI format and converting them to audio files with the extension of wav or pianoroll or music score notation by communicating with other sub-modules.

3. Test and validation of the proposed model using a pre-existing dataset and compare it with other models from the literature.

   Test and validation of the model were evaluated using a non-parametric experiment as presented in section 4.5.2, where a discriminant model using Decision Trees (DT) was used as an internal system to measure the effects of the performance markers in the synthesis process. Also, the pitch, step and duration of the notes were used for external validation.

   Using this paradigm of evaluation the proposed model has an average of 11.33% of better note sequence generation as presented in figure 4.5. By comparison, recurrent neural network has an average of 1.22% of better note sequence generation as presented in figure 4.6.

   The proposed model suggests evidence to formulate new hypothesis regarding the synthesis process using deep neural networks. As a consequence of trying to articulate the expressiveness in music as an operational concept by measuring the effects over a discriminant model to test the results of a generative model.

6 Conclusion and Future Work

In this thesis, the methodology and design of experiments proposed to study the music generation within the framework of machine learning. Essentially, focused on the problem of finding good representation of the data to simplify the learning and testing all assumptions and hypothesis with systematic procedures.

The proposed system for audio generation uses a processing workflow using sequences of an audio file by applying a slide window of 30s with an overlap of 2s. Thus, by using temporal and frequency transformations like zero crossing rate, spectral contrast, spectral centroid and tonal centroid features, a expressive descriptor metric can be generated to improve the identification or separability of different styles or variations of a group that can be defined as a musical genre. This metric can be compacted as vector or matrix of features to reduce the training process of machine learning algorithms.

Results of using the processing workflow with these features shows that the classification accuracy and kappa of machine learning algorithms can be improved up to 70% and 0.6 respectively, even using unbalanced datasets. These results provide evidence of the utility of the selected features for the identification and extraction of audio features related to expressiveness in music over audio files.

Simultaneously the system uses deep convolutional networks to retain or learn patterns of sequences of notes from an audio file. This suggests an advantage over traditional generative models, by processing the audio files as sequences, vectors or matrix, allowing to simplify the parallelization of tasks related to the learning process of machine learning algorithms. This allows to synthesize music audio files on CPU machines and reach similar speeds in comparison to running the model using GPU.

Results suggest that using the proposed metric as measurement of expressiveness in music over discriminative models can improve the quality of the next note in a sequence by 6.5% overall in generative models. In contrast with the results obtained by using other generative algorithms like RNN.

Adding external metrics such as pitch, step and duration of the notes allows observing that the model proposed generates music with the same style by comparing the variance of the music notes using the F-test and having a p-value of 0.00, implying a strong rejection of the null hypothesis of no differences in the pitch range distribution across the different synthetic samples. Additionally, the model can work with audio files using the wav and midi extension, piano roll and score notation to represent the data.

The objectives set out in this thesis were successfully developed and implemented to build a Music system capable of analyzing music recordings, extracting signal level features to create an expressive descriptor, and synthesizing or modeling similar music to the audio samples available.

6.1 Future Work

Future work might be related to testing the methodology using other genres besides classical music. Additionally, find how the transformations of the time or frequency domain affect the classification performance of the discriminative model and its relationship with the synthesis model.

Additionally, we proposed to create a parametric experimental design with the support of musicians and experts to evaluate the results using a mean opinion score over the similarities between the synthetic samples against the original samples.

Alternatively, we propose to train the synthesis model using a set of rules using the explicit form using local variations of each genre with the supervision of an expert and test the performance using the expressive descriptor..

References

[1] Martin Abadi, Paul Barham, Jianmin Chen, et al. “Tensorflow: A system for large-scale machine learning”. In: *12th Symposium on Operating Systems Design and Implementation*. 2016, pp. 265–283 (zitiert auf Seite 38).

[2] Yash Ahuja and Sumit Kumar Yadav. “Multiclass Classification and Support Vector Machine By Yashima Ahuja &”. In: 2012 (zitiert auf Seite 16).

[3] Vincent Akkermans, J. Serrà, and Perfecto Herrera. “Shape-based spectral contrast descriptor”. In: *Sound and Music Computing Conference*. Available online at. Porto, Portugal., 25/07/2009 2009, pp. 143–148 (zitiert auf Seite 26).

[4] Ron Artstein and Massimo Poesio. “Inter-Coder Agreement for Computational Linguistics”. In: *Computational Linguistics* 34.4 (Dec. 2008), pp. 555–596. eprint: `https://direct.mit.edu/coli/article-pdf/34/4/555/1808947/coli.07-034-r2.pdf` (zitiert auf Seite 29).

[5] Sumit Kumar Banchhor and Arif Khan. “Article: Musical Instrument Recognition using Zero Crossing Rate and Short-time Energy”. In: *International Journal of Applied Information Systems* 1.3 (Feb. 2012). Published by Foundation of Computer Science, New York, USA, pp. 16–19 (zitiert auf Seite 26).

[6] Jan Beran. *Statistics in Musicology*. Chapman & Hall/CRC interdisciplinary statistics series. Taylor & Francis, 2004 (zitiert auf den Seiten 6, 7).

[7] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006 (zitiert auf den Seiten 18, 19).

[8] Dibya Jyoti Bora and Anil Kumar Gupta. “Effect of Different Distance Measures on the Performance of K-Means Algorithm: An Experimental Study in Matlab”. In: *CoRR* abs/1405.7471 (2014). arXiv: `1405.7471` (zitiert auf Seite 14).

[9] Koupidis K. Bratsas C., Salanova J-M., Giannakopoulos K., Kaloudis A., and Aifadopoulou G. “A Comparison of Machine Learning Methods for the Prediction of Traffic Speed in Urban Places”. In: *Sustainability* (2020), pp. 12–142 (zitiert auf Seite 1).

[10] 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”. In: *Mach. Learn.* 106.6 (June 2017), pp. 887–909 (zitiert auf den Seiten 1, 13).

[11] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. “Convolutional recurrent neural networks for music classification”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017, pp. 2392–2396 (zitiert auf Seite 19).

[12] Smita Chormunge and Sudarson Jena. “Efficiency and Effectiveness of Clustering Algorithms for High Dimensional Data”. In: *International Journal of Computer Applications* 125 (2015), pp. 35–40 (zitiert auf den Seiten 14, 30).

[13] Jacob Cohen. “A Coefficient of Agreement for Nominal Scales”. In: *Educational and Psychological Measurement* 20.1 (1960), pp. 37–46 (zitiert auf Seite 29).

[14] Yandre M.G. Costa, Luiz S. Oliveira, and Carlos N. Silla. “An evaluation of Convolutional Neural Networks for music classification using spectrograms”. In: *Applied Soft Computing* 52 (2017), pp. 28–38 (zitiert auf Seite 19).

[15] Kwetishe Joro Danjuma. “Performance Evaluation of Machine Learning Algorithms in Post-operative Life Expectancy in the Lung Cancer Patients”. In: *CoRR* abs/1504.04646 (2015). arXiv: `1504.04646` (zitiert auf Seite 1).

[16] Ian Davidson and S. S. Ravi. “Using Instance-Level Constraints in Agglomerative Hierarchical Clustering: Theoretical and Empirical Results”. In: *Data Min. Knowl. Discov.* 18.2 (Apr. 2009), pp. 257–282 (zitiert auf Seite 14).

[17] Giovanni De Poli. “Methodologies for Expressiveness Modelling of and for Music Performance”. In: *Journal of New Music Research* 33 (Sept. 2004), pp. 189–202 (zitiert auf den Seiten 9, 20, 46).

[18] P. Dhanalakshmi, S. Palanivel, and V. Ramalingam. “Classification of audio signals using SVM and RBFNN”. In: *Expert Systems with Applications* 36.3, Part 2 (2009), pp. 6069–6075 (zitiert auf Seite 16).

[19] Matthias Dorfer, Jan Hajic, Andreas Arzt, Harald Frostel, and Gerhard Widmer. “Learning Audio-Sheet Music Correspondences for Cross-Modal Retrieval and Piece Identification”. In: *Trans. Int. Soc. Music. Inf. Retr.* 1 (2018), pp. 22–31 (zitiert auf Seite 25).

[20] Anders Friberg, Roberto Bresin, and Johan Sundberg. “Overview of the KTH rule system for musical performance”. In: *Advances in Cognitive Psychology* 2 (Jan. 2006) (zitiert auf Seite 46).

[21] Claudio Gambella, Bissan Ghaddar, and Joe Naoum-Sawaya. “Optimization problems for machine learning: A survey”. In: *European Journal of Operational Research* 290 (May 2021), pp. 807–828 (zitiert auf Seite 1).

[22] Emden R. Gansner and Stephen C. North. “An open graph visualization system and its applications to software engineering”. In: *SOFTWARE - PRACTICE AND EXPERIENCE* 30.11 (2000), pp. 1203–1233 (zitiert auf Seite 38).

[23] Elkin García, Jorge Pacheco, and Andrés Mancera. “Clasificación de Música por Género Utilizando Redes Neuronales Artificiales”. In: *CIIC*. 2005 (zitiert auf Seite 18).

[24] Patrick Georges. “Western classical music development: a statistical analysis of composers similarity, differentiation and evolution”. In: *Scientometrics* 112 (July 2017), pp. 21–53 (zitiert auf den Seiten 1, 13).

[25] Theodoros Giannakopoulos. “pyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis”. In: *PloS one* 10.12 (2015) (zitiert auf den Seiten 23, 47).

[26] Werner Goebl and Gerhard Widmer. “On the use of computational methods for expressive music performance”. In: Jan. 2009, pp. 93–113 (zitiert auf Seite 46).

[27] Izaro Goienetxea, José María Martínez-Otzeta, Basilio Sierra, and Iñigo Mendialdua. “Towards the use of similarity distances to music genre classification: A comparative study”. In: *PLOS ONE* 13.2 (Feb. 2018), pp. 1–18 (zitiert auf Seite 16).

[28] Maarten Grachten and Florian Krebs. “An Assessment of Learned Score Features for Modeling Expressive Dynamics in Music”. In: *IEEE Transactions on Multimedia* 16.5 (2014), pp. 1211–1218 (zitiert auf Seite 13).

[29] Maarten Grachten and Gerhard Widmer. “Linear Basis Models for Prediction and Analysis of Musical Expression”. In: *Journal of New Music Research* 41.4 (2012), pp. 311–322. eprint: `https://doi.org/10.1080/09298215.2012.731071` (zitiert auf den Seiten 11, 12).

[30] Christopher Harte, Mark Sandler, and Martin Gasser. “Detecting Harmonic Change in Musical Audio”. In: *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia*. AMCMM ’06. Santa Barbara, California, USA: Association for Computing Machinery, 2006, pp. 21–26 (zitiert auf Seite 27).

[31] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. 2nd ed. Springer, 2009 (zitiert auf Seite 17).

[32] Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, et al. “Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset”. In: *CoRR* abs/1810.12247 (2018). arXiv: `1810.12247` (zitiert auf den Seiten 25, 44).

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778 (zitiert auf Seite 35).

[34] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. “Support vector machines”. In: *IEEE Intelligent Systems and their Applications* 13.4 (1998), pp. 18–28 (zitiert auf Seite 16).

[35] Simon Holland. “Artificial Intelligence in music education: a critical review”. In: *Contemporary Music Studies* 20 (Jan. 2000), p. 21 (zitiert auf Seite 2).

[36] ITU-T. “Pulse Code Modulation (PCM) of voice frequencies”. In: *Recommendation G.711* (1988) (zitiert auf Seite 38).

[37] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014 (zitiert auf Seite 7).

[38] Dan-Ning Jiang, Lie Lu, Hong-Jiang Zhang, Jian-Hua Tao, and Lian-Hong Cai. “Music type classification by spectral contrast feature”. In: *Proceedings. IEEE International Conference on Multimedia and Expo*. Vol. 1. 2002, 113–116 vol.1 (zitiert auf Seite 27).

[39] A. Jimenez M. Mauro and Lopez M. Juan S. “System for the measurement of musical similarity, using expressive markers considering the acoustic intensity and temporal metrics”. In: (2018) (zitiert auf Seite 29).

[40] William Strunk Jr. and E. B. White. *The Elements of Style*. Pearson; 4th edición (23th July, 1999), 2003 (zitiert auf Seite 17).

[41] Alexis Kirke and Eduardo Reck Miranda. “A survey of computer systems for expressive music performance”. In: *ACM Comput. Surv.* 42 (2009), 3:1–3:41 (zitiert auf Seite 1).

[42] Daniel Kostrzewa, Robert Brzeski, and Maciej Kubanski. “The Classification of Music by the Genre Using the KNN Classifier”. In: *Beyond Databases, Architectures and Structures. Facing the Challenges of Data Proliferation and Growing Variety*. Ed. by Stanisław Kozielski, Dariusz Mrozek, Paweł Kasprowski, Bożena Małysiak-Mrozek, and Daniel Kostrzewa. Cham: Springer International Publishing, 2018, pp. 233–242 (zitiert auf Seite 16).

[43]Alexandre Lacoste and Douglas Eck. “A Supervised Classification Algorithm for Note Onset Detection”. In: *EURASIP J. Adv. Signal Process* 2007.1 (Jan. 2007), p. 153 (zitiert auf Seite 18).

[44]Yann Lecun and Yoshua Bengio. “Convolutional networks for images, speech, and time-series”. English (US). In: *The handbook of brain theory and neural networks*. Ed. by M.A. Arbib. MIT Press, 1995 (zitiert auf Seite 19).

[45]Jeong-Ran Lee and Hee-Seok Oh. “Circular Statistics in Musicology”. In: 2008 (zitiert auf Seite 6).

[46]Ziming Li and Maarten de Rijke. “The Impact of Linkage Methods in Hierarchical Clustering for Active Learning to Rank”. In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’17. Shinjuku, Tokyo, Japan: Association for Computing Machinery, 2017, pp. 941–944 (zitiert auf Seite 14).

[47]Ghassemi M., Naumann T., Schulam P., et al. “A Review of Challenges and Opportunities in Machine Learning for Health”. In: (2020) (zitiert auf Seite 2).

[48]Brian McFee, Alexandros Metsai, Matt McVicar, et al. *librosa/librosa: 0.8.1rc2*. Version 0.8.1rc2. May 2021 (zitiert auf den Seiten 23, 47).

[49]Anders Meng, Peter Ahrendt, Jan Larsen, and Lars Kai Hansen. “Temporal Feature Integration for Music Genre Classification”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 15.5 (2007), pp. 1654–1664 (zitiert auf Seite 28).

[50]L. Mion and G. De Poli. “Score-Independent Audio Features for Description of Music Expression”. In: *Trans. Audio, Speech and Lang. Proc.* 16.2 (Feb. 2008), pp. 458–466 (zitiert auf Seite 26).

[51]Robert A. Moog and Thomas L. Rhea. “Evolution of the Keyboard Interface: The Bosendorfer 290 SE Recording Piano and the Moog Multiply-Touch-Sensitive Keyboards”. In: *Computer Music Journal* 14 (1990), p. 52 (zitiert auf Seite 24).

[52]F. Morchen, A. Ultsch, M. Thies, and I. Lohken. “Modeling timbre distance with temporal statistics from polyphonic music”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 14.1 (2006), pp. 81–90 (zitiert auf Seite 29).

[53]*Music21 documentation* (zitiert auf Seite 40).

[54]Joshua Neumann. “Music Data Analysis: Foundations and Applications. Ed. by Claus Weihs, Dieter Jannach, Igor Vatolkin, and Guenter Rudolph”. In: *Music and Letters* 99.3 (Dec. 2018), pp. 502–504. eprint: `https://academic.oup.com/ml/article-pdf/99/3/502/27227661/gcy060.pdf` (zitiert auf den Seiten 26, 27).

[55]Stavros Ntalampiras, Ilyas Potamitis, and Nikos Fakotakis. “Exploiting temporal feature integration for generalized sound recognition”. In: *EURASIP J. Adv. Signal Process.* 2009.1 (Dec. 2009) (zitiert auf Seite 29).

[56]Aaron van den Oord, Sander Dieleman, Heiga Zen, et al. *WaveNet: A Generative Model for Raw Audio*. 2016. arXiv: `1609.03499 [cs.SD]` (zitiert auf den Seiten 35, 36, 38).

[57]Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. *Pixel Recurrent Neural Networks*. 2016. arXiv: `1601.06759 [cs.CV]` (zitiert auf Seite 35).

[58] Aäron van den Oord, Nal Kalchbrenner, Oriol Vinyals, et al. “Conditional Image Generation with PixelCNN Decoders”. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS’16. Barcelona, Spain: Curran Associates Inc., 2016, pp. 4797–4805 (zitiert auf Seite 35).

[59] Rui Pedro Paiva, Teresa Mendes, and Amílcar Cardoso. “Melody Detection in Polyphonic Musical Signals: Exploiting Perceptual Rules, Note Salience, and Melodic Smoothness”. In: *Computer Music Journal* 30 (2006), pp. 80–98 (zitiert auf Seite 7).

[60] Athanase Papadopoulos. “Mathematics and group theory in music”. In: *arXiv: History and Overview* 32 (2014) (zitiert auf Seite 6).

[61] F. Pedregosa, G. Varoquaux, A. Gramfort, et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830 (zitiert auf den Seiten 14, 16, 17).

[62] International Federation of the Phonographic Industry (IFPI). *Global Music Report*. Sept. 2020 (zitiert auf Seite 2).

[63] Payam Refaeilzadeh, Lei Tang, and Huan Liu. “Cross-Validation”. In: *Encyclopedia of Database Systems*. Ed. by LING LIU and M. TAMER ÖZSU. Boston, MA: Springer US, 2009, pp. 532–538 (zitiert auf Seite 34).

[64] N. Ruiz Reyes, P. Vera Candeas, S. García Galán, and J.E. Muñoz. “Two-stage cascaded classification approach based on genetic fuzzy learning for speech/music discrimination”. In: *Engineering Applications of Artificial Intelligence* 23 (2010), pp. 151–159 (zitiert auf Seite 1).

[65] Carlos Fernández Collado Roberto Hernandez Sampieri and María del Pilar Baptista Lucio. *Metodología de la Investigación*. McGRAW-HILL and INTERAMERICANA EDITORES, S.A. DE C.V., 2014 (zitiert auf Seite 41).

[66] Lior Rokach and Oded Maimon. *Data Mining With Decision Trees: Theory and Applications*. 2nd. USA: World Scientific Publishing Co., Inc., 2014 (zitiert auf Seite 17).

[67] Justin Salamon and Emilia Gomez. “Melody Extraction From Polyphonic Music Signals Using Pitch Contour Characteristics”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.6 (2012), pp. 1759–1770 (zitiert auf den Seiten 7, 40).

[68] Lukas Schott, Jonas Rauber, Matthias Bethge, and Wieland Brendel. “Towards the first adversarially robust neural network model on MNIST”. In: *Seventh International Conference on Learning Representations (ICLR 2019)*. 2018 (zitiert auf Seite 46).

[69] M. Suchecki and T. Trzciski. “Understanding aesthetics in photography using deep convolutional neural networks”. In: (2017), pp. 149–153 (zitiert auf Seite 1).

[70] João Paulo Teixeira, Paula Odete Fernandes, and Nuno Alves. “Vocal Acoustic Analysis – Classification of Dysphonic Voices with Artificial Neural Networks”. In: *Procedia Computer Science* 121 (2017), pp. 19–26 (zitiert auf Seite 1).

[71] Alaa Tharwat, Tarek Gaber, Abdelhameed Ibrahim, and Aboul Ella Hassanien. “Linear discriminant analysis: A detailed tutorial”. In: *AI Commun.* 30 (2017), pp. 169–190 (zitiert auf Seite 15).

[72] Tippaya Thinsungnoen, Nuntawut Kaoungku, Pongsakorn Durongdumronchai, Kittisak Kerdprasop, and Nittaya Kerdprasop. “The Clustering Validity with Silhouette and Sum of Squared Errors”. In: 2015 (zitiert auf Seite 23).

[73] *Timidity++* (zitiert auf Seite 40).

[74] G. Tzanetakis and P. Cook. “Musical genre classification of audio signals”. In: *IEEE Transactions on Speech and Audio Processing* 10.5 (2002), pp. 293–302 (zitiert auf Seite 28).

[75] T. Vafeiadis, K.I. Diamantaras, G. Sarigiannidis, and K.Ch. Chatzisavvas. “A comparison of machine learning techniques for customer churn prediction”. In: *Simulation Modelling Practice and Theory* 55 (2015), pp. 1–9 (zitiert auf Seite 1).

[76] Sabine Schulte im Walde. “Experiments on the Automatic Induction of German Semantic Verb Classes”. In: *Computational Linguistics* 32.2 (June 2006), pp. 159–194. eprint: `https://direct.mit.edu/coli/article-pdf/32/2/159/1798264/coli.2006.32.2.159.pdf` (zitiert auf Seite 30).

[77] Gerhard Widmer. “Machine Discoveries: A Few Simple, Robust Local Expression Principles”. In: *Journal of New Music Research* 31 (Mar. 2002), pp. 37– (zitiert auf den Seiten 11, 13).

[78] Gerhard Widmer. “Discovering simple rules in complex data: A meta-learning algorithm and some surprising musical discoveries”. In: *Artificial Intelligence* 146.2 (2003), pp. 129–148 (zitiert auf Seite 11).

[79] Gerhard Widmer and Werner Goebl. “Computational Models of Expressive Music Performance: The State of the Art”. In: *Journal of New Music Research* 33 (2004), pp. 203–216. eprint: `https://doi.org/10.1080/0929821042000317804` (zitiert auf den Seiten 1, 9, 11).

[80] S. Wu, S. Zhong, and Y. Liu. “Steganalysis via Deep Residual Network”. In: (2016), pp. 1233–1236 (zitiert auf Seite 1).

Declaration

I, the undersigned, declare that the work I have produced is original and it has been completed only with the help of the references mentioned.

Barranquilla, November 18, 2022

Mauro Alejandro Jimenez
Medina