"Bringing musicke into the tabliture": Machine learning models for polyphonic transcription of lute tablature

Reinier de Valk and Tillman Weyde
Music Informatics Research Group
City University London

1. INTRODUCTION

The historical importance of the polyphonic lute, whose heyday extended from the early sixteenth to the late eighteenth century, cannot be overestimated. Its contemporary popularity is comparable to that of the piano in the nineteenth century or the guitar in the twentieth, and the instrument has left us with a considerable corpus of polyphonic music containing a wide variety of genres: over 860 print and manuscript sources have survived, containing approximately 60,000 pieces (Ness & Kolczynski, 2001). Yet, apart from a number of specialist studies, this music has so far escaped systematic musicological research. The main reason behind this, as also argued by Griffiths (2002) and Rhodes and Lewis (2006), among others, is its notational format—or rather, the (in our modern eyes) *incompleteness* thereof. Lute music was written exclusively in tablature, a succinct, practical notation that instructs the player where to place the left-hand fingers on the fretboard and which strings to pluck with the right hand (see Figure 1). Three points can be identified that lead to tablature's notational succinctness. First, it provides no direct information on pitch, which depends on the tuning of the instrument (different standard tunings were in use). Second, it provides only limited rhythmic information—namely one duration per chord, whose individual notes then can have the given duration, but also be longer or shorter. Third, it does not specify to which polyphonic voice the notes belong. It is because of the latter two points that tablature at first sight reveals very little about the polyphonic structure of the music it encodes (compare Figures 1 and 2). Exactly this notational lacuna, which yields a lot of flexibility, but the bridging of which also presumes a fairly intimate knowledge of both the instrument and the various musical styles it was used for, makes music written in tablature relatively inaccessible to non-specialists. The dearth of its notational format, in other words, thus contributes to what has been described by Griffiths as the "peripheral position of the lute in modern scholarly consciousness" (2002, p. 91).

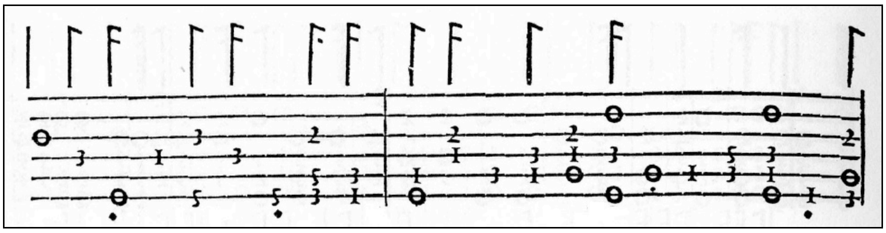

Figure 1. Excerpt of lute tablature in French style.

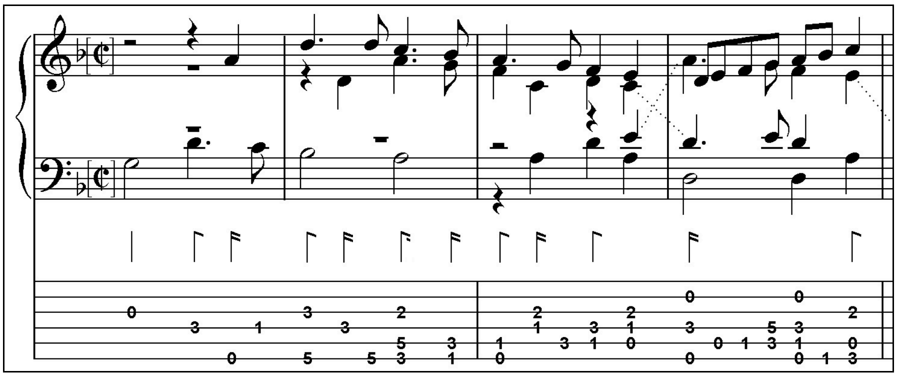

Figure 2. Interpretation in modern music notation.

Increased availability of this music in modern music notation, a format much more familiar to the modern-day musician or researcher, can help unlock the corpus to a larger audience. For the reasons explained above, however, transcribing tablature can be a time-consuming and specialist enterprise. A solution may therefore be sought in *automatic* polyphonic transcription. In this study, we present four variants of a machine learning model—all of them based on neural networks—for voice separation and duration reconstruction in lute tablature.[1] 'Voice separation' is a term used in the field of Music Information Retrieval (MIR), and is defined by Cambouropoulos (2008) as "the task of separating a musical work consisting of multi-note sonorities into independent constituent voices" (p. 75). Machine learning, a branch of Artificial Intelligence, is a discipline in which models are developed that are trained on example data to improve their performance of the task at hand on new, unseen input. In our case, the primary task is voice assignment—that is, to predict voices for notes. However, because note duration plays a significant role in this process, we also experiment with models that predict, for each note, a voice and a duration simultaneously. By means of these models, we hope to bring "musicke into the tabliture": to reconstruct the voice and duration information that is lacking, thus making significant progress towards automatic polyphonic transcription.[2] Secondly, we hope that such models can help us to understand music notated in lute tablature better.

2. RELATED WORK

Up until now, hardly any research on voice separation in lute tablature or automatic polyphonic transcription thereof has been conducted. We have been able to identify only a single related research project, founded as early as 1973 and terminated in the first half of the 1990s (Charnassé & Stepien, 1986; 1992). This project, named ERATTO (Équipe de Recherche sur l'Analyse et Transcription des Tablatures par Ordinateur), focused specifically on automatic transcription of *German* lute tablature—a choice motivated by the observation that this tablature style is the most "abstract in its presentation" (1992, p. 144). In a summarising article (1992), the authors describe two approaches to automatic transcription, both of them rule-based. The first is a 'one-pass restructuring exploration' in which 'saturated chords' serve as anchor points to connect notes to. The connecting process is guided by the principle of continuity, which dictates that voices tend to move in small intervals; ambiguous situations are solved with additional rules and heuristics. The second approach is a 'fixed-rules inference system' in Prolog, consisting of two stages: (i) identification of imitative entries, and (ii) grouping of the notes in chains based on pitch proximity, and connecting the

[1] Although the models have been designed principally for lute tablature, with some minor modifications they can also be applied to other polyphonic music corpora in symbolic format.

[2] William Barley, *A nevv Booke of Tabliture* (London, 1596), excerpt from the "two and twentieth Rule" in the introductory instructions on how to play the lute.

chains to a saturated chord or to one another using several production rules. This approach is reported to give better results, but it is noted that (i) the system is too dependent on the concept of saturated chords, and that (ii) the results may improve when more principles or rules are modelled.

More research has been published on voice separation in non-tablature music formats. Two phases can be discerned. In the first phase (1980s-1990s), the research centred on the modelling of perceptual phenomena relating to polyphonic structure. Huron (1989) presents a model for measuring pseudo-polyphony; Marsden (1992) describes six rule-based models (four of them analogous to neural networks) for modelling the perception of voices in polyphonic music; Gjerdingen (1994) uses a neural network model to model the perception of 'apparent motion', and McCabe and Denham (1997) present a leaky integrator neuron model of the early stages of the process of auditory streaming.

The first phase can be regarded as a preliminary phase or theoretical background for the second phase (2000s), where we see the development of models for voice separation. Two categories can be discerned: rule-based models (Cambouropoulos, 2000; Temperley, 2001; Kilian & Hoos, 2002; Chew & Wu, 2005; Madsen & Widmer, 2006; Szeto & Wong, 2006; Karydis et al., 2007; and Rafailidis, Cambouropoulos, & Manolopoulos, 2009) and machine learning models (Kirlin and Utgoff, 2005; Jordanous, 2008).[3] A detailed discussion of all these models goes beyond the scope of this article; suffice it to say that each of them is based on at least one of two, and sometimes more, fundamental perceptual principles that group notes into voices. These two principles are presented by Huron (2001) as the Pitch Proximity Principle and the Temporal Continuity Principle, and they imply that the closer notes are to one another in terms of pitch or time, respectively, the more likely they are perceived as belonging to the same voice.

3. MODELLING APPROACH AND MODELS

3.1 Processing approaches

We propose two different processing approaches: (i) a forward approach, where the music is processed from left to right, and where information from back in time is used for the voice (and duration) prediction of each note; and (ii) a backward approach, where the music is processed in the opposite direction and the predictions are based on information from ahead in time. In both approaches, the music is processed note by note, always starting with the lowest note in each chord.[4] The forward approach is the most intuitive, as left-to-right is the direction in which music unfolds—new notes get their meaning in the context of what was heard before—and in which we experience it as listeners and performers. The validity of the backward approach is perhaps illustrated most clearly in the fact that the polyphonic fabric tends to become saturated (meaning that all voices are active) towards the end of a piece, whereas at the beginning, the texture tends to be thin (one often finds successive single-voice entries, for example). As a consequence, in the opening bars of a piece, it often becomes clear to which voice the opening notes belong only when the other voices enter—that is, when more context is given. Such problems rarely occur at the end of a piece, as generally, in the closing bars all voices are active, and the final chord almost without exception contains a note from each voice (see Figure 3).[5] Using the backward approach is thus hoped to decrease the

[3] We focus on methods for voice separation in *symbolic* formats; methods designed for audio formats are left out of consideration.

[4] This is an arbitrary choice; it may be worthwhile also to implement processing the chords from top to bottom.

[5] This applies to all the pieces in our data set.

possibility of starting ‘on the wrong foot’, and is therefore expected to aid the voice assignment process.[6]

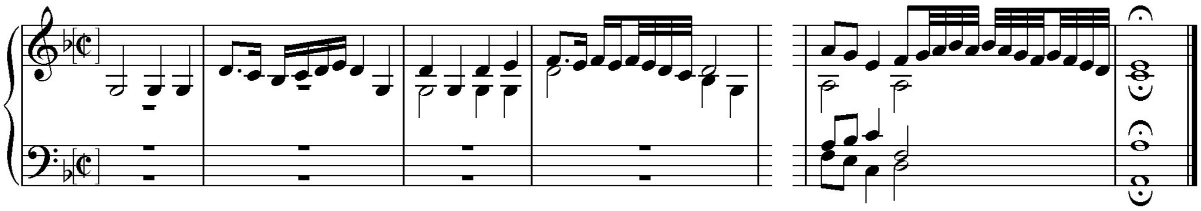

Figure 3. Opening and closing bars of Sebastian Ochsenkun, *Qui habitat* (tablature left out). Note the low tessitura of the Superius’s entry, which in a forward modelling approach is not unlikely to lead to these notes being assigned to a lower voice when the necessary context is lacking.

We experiment with the forward and backward approach (a) modelling only voice, and (b) modelling voice and duration simultaneously, resulting in a total of four models (henceforth referred to as `fwd`; `fwd_dur`; `bwd`; and `bwd_dur`). In the `fwd_dur` model, where both voice and duration information from back in time are used when the voice assignment decision is made, the availability of duration information is expected to facilitate that decision. This is because the voices for all notes that are sustained beyond the onset time of the note for which the decision is made become unavailable (assuming that each voice is considered to be a *monophonic* sequence of notes). Conversely, in the `bwd_dur` model, where voice information from ahead in time is used, the availability of this information is expected to aid the duration prediction. The duration of a note is determined by three (interrelated) factors: (i) the onset time of the next note in the same voice; (ii) the onset time of the next note on the same course, and (iii) whether or not the note has exceeded its (theoretical) maximum duration of a semibreve (or, in modern notation with reduction by half of the values, a half note).[7] In the `bwd_dur` model, all this information is used.[8]

3.2. Models

All our models use a standard three-layer feed-forward neural network with sigmoid activation function and resilient backpropagation (RPROP; Riedmiller & Braun, 1993; improved by Igel & Hüsken, 2000; 2003) as the learning algorithm. This provides a proven fast and robust learning model. The task is modelled as a classification problem, where there are two types of classes: voices and durations. Out of each note in the data set a training example is created: a pair consisting of a feature vector (a numerical representation of the note) and two ground truth labels (one-of-*n* representations of the note’s voice and duration). Given the feature vectors as input, the network is trained (its weights are adapted) so that its

[6] Processing backward in itself is not a novel idea: it has already been suggested and implemented by Charnassé and Stepien (1992), for a similar reason. They state “Th[e] music is heavily embellished, and ornaments usually lead into a chord or sustained note. Consequently, as an ornament approaches its resolution chord, it seems to narrow down its range” (p. 164).

[7] Minamino’s study of sixteenth-century lute treatises on intabulation technique (1988) shows that there was a general concensus among intabulators that a note played on the lute could not be sustained beyond a semibreve (p. 58-59).

[8] Note that both single-pass voice-duration models potentially suffer from a problem arising due to the interrelatedness of voice and duration. In the `fwd_dur` model, duration prediction is harder because voice information ahead in time is lacking, which may lead to less precise voice prediction for notes in the next chord(s). In the `bwd_dur` model, the opposite phenomenon occurs: voice prediction is more difficult because of lacking duration information back in time. This may now lead to less precise duration prediction for notes in the previous chord(s).

output approximates the labels The trained network is then applied to unseen data to predict voices or voices and durations; the predicted labels are determined by the output neurons with the highest activations.

We use batch training, where the network weights are initialised randomly, and where the weight update is performed after each iteration of the learning algorithm over the complete training set. We have established that for all models, 400 iterations are sufficient for convergence.

Although all models use the same three-layer network architecture, the sizes of the layers vary per model. The size of the input layer equals the number of features, and the size of the output layer is determined by the size of the required label. As a voice label we use a five-dimensional binary vector, each element of which represents a voice, and as a duration label a 32-dimensional binary vector, each element of which represents one-thirtysecond of a whole note.[9] When modelling both voice and duration, these vectors are simply concatenated. The size of the hidden layer, lastly, is a hyperparameter that is optimised for each model in a preliminary grid search. A second hyperparameter optimised in the same grid search is the regularisation parameter λ, which controls the weight decay, the degree to which the network weight sizes are penalised during the training.[10] Both are hyperparameters that influence the complexity of a network. A too complex model runs the risk of overfitting the data, resulting in poor generalisation, while a too simple model may underfit, resulting in poor adaptation to the data. Table 1 gives an overview of the layer sizes and parameter settings for each model as determined in the grid search.

Table 1. Layer sizes and parameter settings for all models.

Model	**Layer size**			λ
	input	**output**	**hidden**	
`fwd`	59	5	30	0.00003
`fwd_dur`	59	37	30	0.00001
`bwd`	59	5	59	0.0001
`bwd_dur`	59	37	59	0.0001

The software is written in Java; most of it, notably the modules for reading tablature and feature calculation, from scratch. Where a representation in modern music notation was needed we used the MUSITECH framework (Weyde, 2005), a computational framework for analysing music, and for the machine learning we used the Encog Machine Learning Framework (http://www.heatonresearch.com/encog).

3.3 Features

Feature design is an essential part of the modelling, as the features, which are given as input to the model, must convey information that is relevant for the task. A feature vector is a numerical representation of properties of a note in its polyphonic context. For all models, we

[9] Note that this allows for five voices and a maximum duration of a whole note. Due to physical limitations of the lutenist and the technical constraints of his instrument, five is generally considered to be the highest number of voices that can be rendered simultaneously on a lute. Furthermore, although a half note was considered to be the longest sustainable duration on the lute, longer durations do occur—not only to complete the final bar (see for example Figure 3), but also occasionally in the middle of pieces. A model whose output layer has five neurons allocated to voices and 32 to durations can thus be applied to music with any number of voices smaller or equal to five and any number of durations ranging from a thirtysecond to a whole note; neurons corresponding to non-existing voices or durations will never be activated.

[10] Using cross-validation, we tested 10 different values of λ (0.1, 0.03, 0.01, 0.003, …, 0.00001, and 0.0 or no regularisation) in combination with six different hidden-layer sizes (0.125, 0.25, 0.5, 1, 2, and 4 times the size of the input layer).

use a 59-dimensional feature vector consisting of three types of features: (i) note- or tablature-specific features, requiring only tablature information; (ii) chord-level features requiring tablature and duration information; and (iii) polyphonic embedding features, requiring tablature, duration, and voice information. An overview of all features is given in Tables 2.1-3; further explanation is given below each table.

Table 2.1. Note- or tablature-specific features.

Note- or tablature-specific features		
0	`pitch`	pitch
1	`course`	course (string pair) used to produce the note
2	`fret`	position on the course
3	`minDuration`	duration as indicated in the tablature
4	`maxDuration`	duration as determined by the next note on the course
5	`isOrnamentation`	true (1) if a sixteenth or smaller and the only note in the chord
6	`metricPosition`	metric position within the bar
7, 13, 19	`interOnsetTimeProx`	inter-onset time proximity to next chord *c*
8, 14, 20	`pitchProxToNoteBelow`	pitch proximity to any closest note below in next chord *c*
9, 15, 21	`courseOfNoteBelow`	course of any closest note below in next chord *c*
10, 16, 22	`courseOfSamePitch`	course of any note with the same pitch in next chord *c*
11, 17, 23	`pitchProxToNoteAbove`	pitch proximity to any closest note above in next chord *c*
12, 18, 24	`courseOfNoteAbove`	course of any closest note above in next chord *c*
25	`numNewNotesNext`	the number of notes in the next chord

Features 0-6 are fairly self-explanatory. Pitches are represented by MIDI numbers, durations and onset- as well as offset times are measured in whole notes, and proximities are inverted distances (explained in more detail below). We assume the same tuning (G) for all pieces in our data set, so that the same tablature symbol always corresponds to the same pitch. Features 7-24 and 25 are intended to facilitate the duration prediction and are expected to be of use most in the `fwd_dur` model, where only the onset time of the next note on the same course (one of the factors that determines the duration of a note; see Section 3.2 above) is known. Together with `maxDuration`, these features, which encode pitch- and time proximity as well as course information to the next *n* chords (where we have established empirically that *n* = 3 is a reasonable choice), are the only forward-looking features in the `fwd` and `fwd_dur` models.

Table 2.2. Chord-level features.

Chord-level features		
26	`chordSize`	number of notes in the chord
27	`indexInChord`	index (based on pitch) in the chord
28	`pitchDistToNoteBelow`	pitch distance to the note below in the chord
29	`pitchDistToNoteAbove`	pitch distance to the note above in the chord
30-33	`intervalsInChord`	intervals in the chord

Moving to the chord-level features, it must be noted that pitch distances and intervals in the chord-level features are measured in semitones. Furthermore, when calculating these features in the `fwd_dur` model, all sustained previous notes are considered. `intervalsInChord` is a four-dimensional vector containing the successive intervals in the chord; for each note the chord is short of the maximum number of notes possible (five), an *n/a* value of -1 is added.

(This value is used for all features encoding some relation to another note, such as pitch distances or proximities, when that other note does not exist.)

Table 2.3. Polyphonic embedding features.

Polyphonic embedding features		
34-38	`voicesWithAdjacent NoteOnSameCourse`	binary vector encoding the voices with an adjacent note on the same course
39-43	`pitchProx`	proximities in pitch to the adjacent note in each voice
44-48	`interOnsetProx`	proximities in time (inter-onset) to the adjacent note in each voice
59-53	`offsetOnsetProx`	proximities in time (offset-onset) to the adjacent note in each voice
54-58	`voicesAlreadyAssigned`	binary vector encoding the voices already assigned in the chord

In the polyphonic embedding features, lastly, the position of what is called the 'adjacent note' in Table 2.3 depends on the model used: in the forward models, it is the previous note in the voices; in the backwards model, it is the next note. Proximity, then, is defined as the inverted distance of note *n* to the adjacent note *a* in voice *v*. We discern
(i) pitch proximity:

$$\texttt{pitchProx}(v) = \quad 1 / (\,|p_n - p_a| + 1\,) \tag{1}$$

(ii) inter-onset proximity:

$$\texttt{interOnsetProx}(v) = \begin{cases} \text{if } on_a < on_n\text{: } 1 / (\,(on_n - on_a) + 1\,) & \text{(2a)} \\ \text{if } on_a > on_n\text{: } 1 / (\,(on_n - on_a) - 1\,) & \text{(2b)} \end{cases}$$

and (iii) offset-onset proximity:

$$\texttt{offsetOnsetProx}(v) = \begin{cases} \text{if } off_a \leq on_n\text{: } 1 / (\,(on_n - off_a) + 1\,) & \text{(3a)} \\ \text{if } off_a > on_n\text{: } 1 / (\,(on_n - off_a) - 1\,) & \text{(3b)} \end{cases}$$

where *p* denotes pitch, *on* onset time, and *off* offset time. As far as offset-onset proximity goes, two things must be noted. First, when using the backward models, offset-onset proximity as defined above (which, when negative, indicates that *a* is sustained beyond the onset time of *n*, rendering voice *v* unavailable for *n*) contains no useful information. What is relevant in the backward models is the difference between the offset time of *n* and the onset time of *a*: if the former exceeds the latter, voice *v* becomes unavailable for *n*. Hence, in the backward models offset-onset proximity is defined as:[11]

$$\texttt{offsetOnsetProxBwd}(v) = \begin{cases} \text{if } off_n < on_a\text{: } 1 / (\,(off_n - on_a) - 1\,) & \text{(4a)} \\ \text{if } off_n \geq on_a\text{: } 1 / (\,(off_n - on_a) + 1\,) & \text{(4b)} \end{cases}$$

[11] Note the symmetry between the forward and the backward models: if there is no reason for 'alarm'—that is, if voice *v* is available for *n*—, the sign of `offsetOnsetProx` is the same as the sign of `interOnsetProx`; otherwise, it is the opposite.

Second, when using the `fwd` or `bwd` models (that is, when using a model that does not predict duration), the duration as given in the tablature is used to determine the offset time of *a* and *n*, respectively.

The feature vector is concluded with `voicesAlreadyAssigned`, a binary vector indicating which voices are no longer available. In the `fwd`, `bwd`, and `bwd_dur` models, only the voices assigned to any lower chord notes are included; in the `fwd_dur` model, to this are also added the voices assigned to any sustained previous notes.

The individual feature values can vary considerably: pitch, for example, moves roughly between 40 and 75, whereas the values of the binary or boolean features can only be 0 or 1. Therefore, each feature is scaled so that all individual feature values in the vector fall within a range between (and including) -1 and 1. Each feature is scaled as follows:

$$\texttt{fScaled} = \begin{cases} \text{if } f_{i\_max} \mathrel{!}= f_{i\_min}\text{: } (f_i - f_{i\_min}) \,/\, (f_{i\_max} - f_{i\_min}) \\ \text{if } f_{i\_max} == f_{i\_min}\text{: } 0.0 \end{cases} \tag{5}$$

where f_i is the feature at index *i* in the feature vector; f_{i_min} is the minimum value (over all feature vectors) for f_i; and f_{i_max} is the maximum value (over all feature vectors) for f_i.

4. DATA SET

Our data set contains nine intabulations, all for four voices (the most common intabulation format), and comprises a total of 8892 notes (see Table 3).[12] All these works are instrumental arrangements of polyphonic sacred and secular vocal pieces by renowned composers from the second half of the fifteenth and the first half of the sixteenth century, such as Josquin des Prez, Heinrich Isaac and Clément Janequin. We have aimed to make the data set as heterogeneous as possible, including works from different centres of lute activity, from different decades, arranging different vocal genres, and consisting of different polyphonic textures.[13]

Table 3. The data set.

Intabulation		Model		
Title and source	Notes	Genre	Composer	Texture
Absolon fili mi. Sebastian Ochsenkun, *Tabulaturbuch auff die Lauten* (Heidelberg, 1558)	1184	motet	Josquin	imitative
In exitu Israel de Egipto. Sebastian Ochsenkun, *Tabulaturbuch auff die Lauten* (Heidelberg, 1558)	1974	motet	Josquin	imitative
Qui habitat. Sebastian Ochsenkun, *Tabulaturbuch auff die Lauten* (Heidelberg, 1558)	2238	motet	Josquin	imitative
Herr Gott laß dich erbarmen. Sebastian Ochsenkun, *Tabulaturbuch auff die Lauten* (Heidelberg, 1558)	371	lied	Isaac	free
Bramo morir. Antonio Rotta, *Intabolatvra de lavto, Libro primo* (Venice, 1546)	708	madrigal	Festa/ Arcadelt	free
Tant que uiuray [a4]. Pierre Phalèse (publ.), *Des Chansons Reduictz en Tabulature de Lvt, Liure premier* (Louvain, 1547)	457	chanson	Sermisy	free
Mais mamignone. Julio Abondante, *Intabolatvra di lavtto,*	705	chanson	Janequin	semi-

[12] The term 'intabulation' is used here in its specific meaning denoting an instrumental arrangement of a vocal piece, not simply any work written in tablature.

[13] The term 'imitative' is used here to denote works whose polyphonic structure is governed by motivic imitation; the term 'semi-imitative' to denote works that contain some imitation, but whose structure is not governed by it.

Libro secondo (Venice, 1548)				imitative
Las on peult. Pierre Phalèse (publ.), *Theatrvm mvsicvm* (Louvain, 1563)	777	chanson	Janequin	semi-imitative
Il nest plaisir. Iulio Caesaro Barbetta, *Il terctio libro de intavolatvra de livto* (Strasbourg, 1582)	478	chanson	Janequin	semi-imitative
Total	8892			

The motivation for using intabulations only is threefold. First, they formed the predominant lute genre in the sixteenth century, and thus are highly representative of the contemporary lute practice. Second, since the densest polyphonic structures in lute music are found in intabulations, they constitute a sub-corpus that is challenging for our research. Third, intabulations provide an objective way of labelling the data in order to acquire the ground truth information needed for the machine learning process. Studies of intabulations and the intabulation process (Ward, 1952; Brown, 1973-74, 1976; Thibault, 1976; Göllner, 1984) and studies of contemporary treatises on the subject (Minamino, 1988; Canguilhem, 2001) show that it was generally of great concern to intabulators, who intabulated the vocal models voice for voice, to retain the polyphonic fabric of the model in their arrangement. The voices in the intabulations can thus be reconstructed by aligning them with their models, whose polyphonic structure is unambiguous.

There is, however, a component of subjectivity to the alignment process, which for this study was carried out by hand. This is because intabulations are generally not literal, verbatim transcriptions of their vocal counterparts: not only did the technical constraints of the lute at times necessitate reworkings of otherwise not playable passages, but intabulators sometimes also simply *chose* to change the original notes. Thus, we find added notes (ornamental or other), altered notes (in pitch, duration, or both), omitted notes, and even complete omitted bars or longer sections.[14] Such adaptations can lead to ambiguities when comparing intabulation and model, and because often several interpretations are contrapuntally plausible, they can thus complicate the alignment process.[15]

5. EVALUATION

The models are evaluated using cross-validation, where the data set is divided into *n* folds, and the model is trained on *n*-1 folds and then tested on the remaining fold. We set *n* equal to the number of pieces in the data set, making each piece a fold. This is repeated *n* times until all folds have served as test set once; the error rates, measured on both the training and the test set, are then averaged over all folds. These error rates are determined by comparing, per fold, the predicted voice and duration labels with the ground truth labels, and then calculating the percentages of notes that were misassigned. When averaging, the error rates are weighted by the number of notes, so that averaged values are per-note error rates.

The error rates on the training data inform us about the models' *adaptation* to the data, that is, how well they learn; the error rates on the test data about their *generalisation*, that is, how well they perform the task on unseen data. On the test data, where the predicted labels are created incrementally, we evaluate the trained model in two different modes. In test mode, we calculate the feature vectors using the ground truth labels (the labels are needed for the chord-level and polyphonic embedding features), and we evaluate the predictions based thereon. In application mode, which corresponds to the real-world situation where ground truth information is not provided, we calculate the feature vectors using the labels as predicted *by the network* earlier in the process. In application mode, errors can therefore propagate:

[14] A more detailed overview of such adaptations is given by Minamino (1988, p. 89 ff.).

[15] We are therefore not claiming to have established the only correct mapping of notes to voices (nor do we want to—one of the virtues of tablature is its multi-interpretability); we can say, however, to provide a *valid* mapping.

once a note has been assigned to the wrong voice or given the wrong duration, this will influence the decision process for the assignment of the following notes, as incorrect information is now used for the feature calculation.

6. RESULTS AND DISCUSSION

Table 4 shows the performance of the models on the data set.

Table 4. Performance of the models on the data set. Error rates are weighted averages over all folds.

Model	Voice prediction			Duration prediction		
	Training error (%)	Test error (%)	Application error (%)	Training error (%)	Test error (%)	Application error (%)
`fwd`	8.30	9.80	20.54	n/a	n/a	n/a
`fwd_dur`	3.08	4.20	27.85	26.56	29.59	25.05
`bwd`	5.39	7.48	20.31	n/a	n/a	n/a
`bwd_dur`	5.88	7.63	20.85	18.33	21.32	15.65

With regard to the models' voice prediction performance, several observations can be made. Let us first consider the forward versus the backward modelling approach. In Section 3 we have argued that modelling backward, which benefits from the fact that the polyphonic fabric tends to get saturated towards the end of a piece, might be a more promising approach for voice separation than modelling forward. The results, however, indicate that this is not always so. Looking at the training and test errors first, we notice that, although the two backward models outperform the `fwd` model, the `fwd_dur` model outperforms all the others by far. Second, we notice that the performance of the two backward models is very similar (in all modes), meaning that modelling duration has no noticeable effect on the voice prediction here. This was anticipated, as for voice prediction it is the duration of notes *back* in time (which in the backward approach is not available) that is relevant (see Section 3.1). We can clearly see this being corroborated when looking at the error rates for the forward models, where information on the duration of previous notes is available: in both the training and test error we witness an error rate decrease by more than a half.

The application errors, however, do not confirm the picture given by the training and test errors. In application mode, the `fwd_dur` model performs notably worse than the other three; moreover, the difference between the error rates of the `fwd` model on the one hand and the two backward models on the other, so clearly visible in training and test mode, have now disappeared. An investigation of the models' output shows that this smoothing out can be explained by error propagation—that is, incorrect assignments leading to more incorrect assignments (as explained in Section 5)—, from which all models suffer. The output also gives a strong indication that the lesser performance of the `fwd_dur` model is directly related to its incapacity of predicting duration well (explained in more detail below). As argued in Section 3.1, the availability of duration information when the voice assignment decision is made is expected to facilitate that decision—an assumption that is corroborated by the decrease in training and test errors between the forward models. However, this also means that when this duration information is incorrect, it is likely to influence the voice assignment decision negatively. We thus hypothesise that the `fwd_dur` model's poor capacity to predict duration is what causes the high application error in the `fwd_dur` model.

With regard to duration prediction performance, two observations can be made. First, as reflected by the training, test, and application errors, the backward approach is clearly more suitable for duration prediction. As explained in Section 3.1, this can be ascribed to the availability of voice information ahead in time, which is lacking in the forward approach. Second, we notice that the application error for both approaches is lower than both the training and the test error. The explanation must be sought in duration reassignments due to encountered conflicts that can occur in application mode. Such conflicts arise when (i) a voice is predicted that has already been assigned to a sustained note (fwd_dur model only); and (ii) a duration is predicted that exceeds the onset time of the next note in the predicted voice (bwd_dur model only).[16] In these cases, the duration of the note that is causing the conflict—in case (i) the sustained note and in case (ii) the note for which the decision is being made—is shortened. In both approaches, these adaptations lead to a considerable decrease in duration error rates (of approximately 20.0% in the fwd_dur and 37.0% in the bwd_dur model).[17]

7. CONCLUSION AND FUTURE WORK

In this study we presented and evaluated four neural network models for predicting voices and durations in lute tablature. These models form the heart of a system for automatic polyphonic transcription of lute tablature, which takes a piece in tablature as input and outputs a polyphonic transcription in modern music format. Currently, the system does not handle problems such as pitch spelling, key detection, or meter detection, as they are considered to be of lesser importance for the task of voice separation and duration prediction.[18] As a consequence, visually the system output is still fairly crude (see Figure 4).

Figure 4. Output as given by the MUSITECH framework.

Moreover, the performance of the models varies and is open for improvement. The results from the fwd_dur model, which performs very well on voice prediction in training and test mode, and those from the bwd_dur model, which performs well on voice prediction in training and test mode and on duration prediction in all modes, show that notably these approaches are promising. In application mode, however, the performance of both models drops—in the case of the fwd_dur model considerably. Error propagation is the main cause for this, to which in the case of the fwd_dur model a poor duration prediction capacity is added. These problem are therefore worthwhile investigating further.

Despite these issues, in the current state the system output can already serve as a ‘first suggestion’, to be adapted to the wishes or interpretation of the user. These may very well differ from the system output even if that output is ‘correct’ according to the ground truth, as the interpretation of lute tablature, where often multiple solutions are contrapuntally plausible and where *the* transcription does not exist, can allow for a fair part of leeway. Therefore, an avenue we wish to follow in future work is to make the system interactive by enabling users to adapt the output, and then feeding the user input back into the system and using it, in conjunction with the ground truth, to re-train the models.

Additionally, we plan to experiment with more elaborate modelling approaches. The forward and backward models are limited in the sense that the polyphonic context within which the voice and duration assignment decisions are taken is only one-directional. A

[16] Additional conflicts with regard to voice prediction arise when a voice is predicted that has already been assigned to a lower note in the same chord. In this case the voice assignment for one of these two notes , (depending on the context) is adapted. However, an analysis of the results shows that such conflicts are few and far between; they therefore do not influence the results significantly.

[17] The opposite phenomenon, where durations initially predicted correctly are adapted into incorrect durations also occurs, but this happens far less often.

[18] Several algorithms for such problems exist; a possible solution is to include implementations of these into the system.

possible approach is to combine these models into a bi-directional model, to be applied in a second pass, where the predictions are based on information from both back and ahead in time. The rationale behind such an approach is that modelling a larger decision context extending in both time directions represents the notes' polyphonic embedding better, and may thus improve the voice assignment. Furthermore, using machine learning models for polyphonic transcription enables the investigation of musical parameters such as style, genre, idiom, texture, etc. by contrasting results on data sets that differ with respect to these parameters. Experiments in this direction are therefore planned as well.

There are, in conclusion, many possible avenues to follow. The models presented in this work, even though their performance is still open for improvement, constitute a promising starting point for the suggested line of research.

8. REFERENCES

Brown, H. M. 1976. "Accidentals and ornamentation in sixteenth-century intabulations of Josquin's motets." In *Josquin des Prez: Proceedings of the International Josquin Festival-Conference held at the Juilliard School at Lincoln Center in New York City, 21-25 June 1971*, edited by E. E. Lowinsky and B. J. Blackburn, 475-522. London: Oxford University Press.

Brown, H. M. 1973-1974. "Embellishment in early sixteenth-century Italian intabulations." *Proceedings of the Royal Music Association* 100: 49-83.

Cambouropoulos, E. 2000. "From MIDI to traditional musical notation." *Proceedings of the AAAI Workshop on Artificial Intelligence and Music: Towards Formal Models for Composition, Performance, and Analysis, Seventeenth National Conference on Artificial Intelligence*. Accessed July 27, 2014. http://users.auth.gr/emilios/englishpage/pub.html.

Cambouropoulos, E. 2008. "Voice and stream: Perceptual and computational modeling of voice separation." *Music Perception* 26: 75-94.

Canguilhem, P. 2001. *Fronimo de Vincenzo Galilei*. Paris: Minerve.

Charnassé, H., and Stepien, B. 1992. "Automatic transcription of German lute tablatures: An artificial intelligence application." In *Computer representations and models in music*, edited by A. Marsden, and A. Pople, 143-70. London: Academic Press.

Charnassé, H., and Stepien, B. 1986. "Automatic transcription of sixteenth-century musical notations." *Computers and the Humanities: A Newsletter* 20.3: 179-90.

Chew, E., and Wu, X. 2005. "Separating voices in polyphonic music: A contig mapping approach." In *Computer Music Modeling and Retrieval: Second International Symposium, CMMR 2004, Esbjerg, Denmark, May 2004, revised papers*, edited by U. Kock Wiil, 1-20. Berlin: Springer. Accessed July 27, 2014. doi:10.1007/978-3-540-31807-1_1.

Gjerdingen, R. O. 1994. "Apparent motion in music?" *Music Perception* 11: 335-70. Accessed July 27, 2014. doi:10.2307/40285631.

Göllner, M. L. 1984. "On the process of lute intabulation in the sixteenth century." In *Ars iocundissima: Festschrift für Kurt Dorfmüller zum 60. Geburtstag*, edited by H. Leuchtmann and R. Münster, 83-96. Tutzing: Schneider.

Griffiths, J. 2002. “The lute and the polyphonist.” *Studi Musicali* 31: 89-108.

Huron, D. 1989. “Voice segregation in selected polyphonic keyboard works by Johann Sebastian Bach.” PhD diss., University of Nottingham.

Huron, D. 2001. “Tone and voice: A derivation of the rules of voice-leading from perceptual principles.” *Music Perception* 19: 1-64. Accessed July 27, 2014. doi:10.1525/mp.2001.19.1.1.

Igel, C., and Hüsken, M. 2000. “Improving the Rprop learning algorithm.” In *Proceedings of the Second ICSC Symposium on Neural Computation*, edited by H.-H. Bothe, and R. Rojas, 115–21. Millet, AB, Canada: ICSC Academic Press.

Igel, C., and Hüsken, M. 2003. Empirical evaluation of the improved Rprop learning algorithms. *Neurocomputing* 50: 105-23. Accessed July 27, 2014. doi:10.1016/S0925-2312(01)00700-7.

Jordanous, A. 2008. “Voice separation in polyphonic music: A data-driven approach.” *Proceedings of the International Computer Music Conference*. Accessed July 27, 2014. http://www.sussex.ac.uk/intsys/people/list/person/197392.

Karydis, I., Nanopoulos, A., Papadopoulos, A., Cambouropoulos, E., and Manolopoulos, Y. 2007. “Horizontal and vertical integration/segregation in auditory streaming: A voice separation algorithm for symbolic music data.” *Proceedings of the 4th Sound and Music Computing Conference*, 299-306. Accessed July 27, 2014. http://www.smcnetwork.org.

Kilian, J., and Hoos, H. H. 2002. “Voice separation—A local optimisation approach.” *Proceedings of the 3rd International Conference on Music Information Retrieval*. Accessed July 27, 2014. http://www.ismir.net.

Kirlin, P. B., and Utgoff, P. E. 2005. “VoiSe: Learning to segregate voices in explicit and implicit polyphony.” *Proceedings of the 6th International Conference on Music Information Retrieval*, 552-557. Accessed July 27, 2014. http://www.ismir.net.

McCabe, S. L., and Denham, M. J. 1997. “A model for auditory streaming.” *Journal of the Acoustical Society of America* 101: 1611-21.

Madsen, S. T., and Widmer, G. 2006. “Separating voices in MIDI.” *Proceedings of the 7th International Conference on Music Information Retrieval*. Accessed July 27, 2014. http://www.ismir.net.

Marsden, A. 1992. “Modelling the perception of musical voices: A case study in rule-based systems.” In *Computer representations and models in music*, edited by A. Marsden and A. Pople*,* 239-63. London: Academic Press.

Minamino, H. 1988. “Sixteenth-century lute treatises with emphasis on process and techniques of Intabulation.” PhD diss., University of Chicago.

Ness, A. J., and Kolczynski, C. A. 2001. “Sources of lute music.” In *The new Grove dictionary of music and musicians*, 2nd ed., Vol. 24, edited by S. Sadie, 39-63. London: Macmillan.

Rafailidis, D., Cambouropoulos, E., & Manolopoulos, Y. 2009. “Musical Voice Integration/ Segregation: *VISA* Revisited.” *Proceedings of the 6th Sound and Music Computing Conference*, 42-47. Accessed July 27, 2014. http://www.smcnetwork.org.

Rhodes, C., and Lewis, D. 2006. “An editor for lute tablature.” In *Computer Music Modeling and Retrieval: Third International Symposium, CMMR 2005, Pisa, Italy, September 2005, revised papers*, edited by R. Kronland-Martinet, T. Voinier, and S. Ystad, 259-64. Berlin: Springer. Accessed July 27, 2014. doi:10.1007/11751069_23.

Riedmiller, M., and Braun, H. 1993. “A direct adaptive method for faster backpropagation learning: The RPROP algorithm.” In *1993 IEEE International Conference on Neural Networks: San Francisco, California, March 28-April 1, 1993,* edited by E. H. Ruspini, volume 1, 586-91. Piscataway, NJ: IEEE Press. Accessed July 27, 2014. doi:10.1109/ICNN.1993.298623.

Szeto, W. M., and Wong, M. H. 2006. “Stream segregation algorithm for pattern matching in polyphonic music databases.” *Multimedia Tools and Applications* 30: 109-127. Accessed July 27, 2014. doi:10.1007/s11042-006-0011-9.

Temperley, D. 2001. *The cognition of basic musical structures*. Cambridge, MA: The MIT Press.

Thibault, G. 1976. “Instrumental transcriptions of Josquin’s French chansons.” In *Josquin des Prez: Proceedings of the International Josquin Festival-Conference held at the Juilliard School at Lincoln Center in New York City, 21-25 June 1971*, edited by E. E. Lowinsky and B. J. Blackburn, 455-74. London: Oxford University Press.

Ward, J. 1952. The use of borrowed material in 16th-century instrumental music. *Journal of the American Musicological Society*, 5: 88-98. Accessed July 27, 2014. doi:10.2307/830181.

Weyde, T. 2005. “Modelling cognitive and analytic musical structures in the Musitech framework.” In *Proceedings of the 5th Conference on Understanding and Creating Music.* Accessed July 27, 2014. http://www.soi.city.ac.uk/~sa746/?cont=1.