Modeling Bends in Popular Music Guitar Tablatures

Alexandre D'Hooge, Louis Bigo, Ken Déguernel

MODELING BENDS IN POPULAR MUSIC GUITAR TABLATURES

Alexandre D'Hooge **Louis Bigo** **Ken Déguernel**
Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
alexandre.dhooge@algomus.fr

ABSTRACT

Tablature notation is widely used in popular music to transcribe and share guitar musical content. As a complement to standard score notation, tablatures transcribe performance gesture information including finger positions and a variety of guitar-specific playing techniques such as *slides*, *hammer-on/pull-off* or *bends*. This paper focuses on bends, which enable to progressively shift the pitch of a note, therefore circumventing physical limitations of the discrete fretted fingerboard. In this paper, we propose a set of 25 high-level features, computed for each note of the tablature, to study how bend occurrences can be predicted from their past and future short-term context. Experiments are performed on a corpus of 932 *lead guitar* tablatures of popular music and show that a decision tree successfully predicts bend occurrences with an F_1 score of 0.71 and a limited amount of false positive predictions, demonstrating promising applications to assist the arrangement of non-guitar music into guitar tablatures.

1. INTRODUCTION

The guitar, whether acoustic or electric, is (in most cases) a fretted instrument which enforces the playing of discrete pitch values on a chromatic scale. This constraint can be beneficial, as it limits the risk of playing out-of-tune. However, it also prevents microtonal experiments or continuous pitch shifts, which can be a powerful means of musical expressiveness. To overcome this limitation, guitarists can alter the string tension with their fretting hand [1] to reach a completely new pitch, up to several semitones higher. This technique is called string bending, or just *bends,* and is an important part of guitar playing in blues, rock or pop music. Even though bending a string can only increase the pitch of a note, a variety of bend types are used. While guitar players mostly agree over the existing variations, the names used can differ. In this paper, we consider the five bend types described in Gomez's work [2]: *Basic upward*, *Held*, *Reverse*, *Up & Down*, and *Complex* bends for bends that do not belong to any of the previous categories.

Guitar tablatures, compared to standard staff notation, include fingering information on where to play a note with a given pitch on the fretboard. Tablatures are therefore an effective notation to display playing techniques, the position of the fretting hand being critical to know how to perform a bend. Examples of bends are shown in a tablature in Figure 1, different bend types are represented with differently shaped arrows. Knowing where and when to use bends is part of the idioms of guitar pop music and an important part of learning this style. However, the variety of bend types can make it difficult to choose how to use them. For instance, a guitarist playing a score that was composed for another instrument may want to add expressiveness using bends, and could need help on deciding which notes to bend and which bend to use. Moreover, a tool suggesting bends could also improve the quality of online tablatures that sometimes do not have guitar techniques annotations.

Could bends be inferred from musical context? Are they correlated to other elements of a score or a tablature such as pitch, rhythm, or hand position? In this paper, we propose to model bent notes and their context through temporal, pitch and tablature related information. Such a representation could be used to predict which notes are bent from a score or a tablature. Our contribution is three-fold: (1) we define a set of high-level features to model bent notes and their context, (2) we conduct a statistical study on bends based on those features, and (3) we propose a method for predicting bends from tablatures.

The rest of this paper is organized as follows: after discussing related works in Section 2, we introduce our modeling choices in Section 3. The dataset and its statistical study are presented in Section 4. We introduce our prediction algorithm in Section 5 and reflect on this work in Section 6.

2. RELATED WORKS

Guitar tablatures are often studied in the audio realm for guitar music transcription [3, 4]. In particular, automatic transcription of playing techniques from audio has also been studied, for instance with hexaphonic microphones [5] or deep learning techniques [6]. Of course, this task is not restricted to guitar and has been applied to other instruments such as the Chinese *guqin*, which also features several string bending techniques [7].

Another related task is the generation of tablatures, which has been studied increasingly in the last decade. Playing techniques can be included in generation frameworks to obtain results closer to actual human performance. The Transformer-XL model presented in [8, 9] can for instance generate tokens representing playing tech-

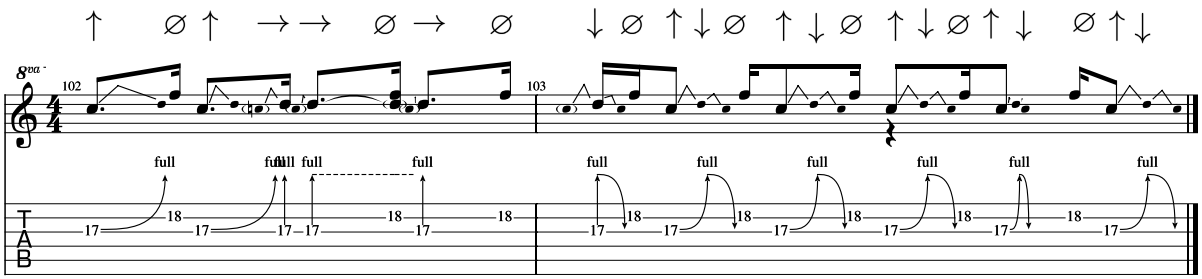


Figure 1: Excerpt from Lynyrd Skynyrd's *Free Bird* solo. In the first measure, the first two bends are *basic upward bends*, the remaining ones are *held bends*. In the second measure, the first bend is a *reverse bend*, and the other ones are *up and down bends*. While all bends of this example are whole tones (denoted by *full*), the amplitude of a bend can vary. String movements are labeled above, according to the representation presented in Section 3. An audio rendering of this excerpt is available on the **accompanying repository**.

niques. Chen et al. [10] also consider *Technique* events, but only for picking-hand techniques, since the generation focuses on fingerstyle guitar. McVicar et al. present in [11] a method to generate tablatures of guitar solo phrases, and fretting-hand techniques are added in a post-processing step. Beyond the case of guitar, playing techniques modeling includes symbolic piano music generation with sustain pedal information [12].

Another large part of guitar tablature MIR research focuses on fingering prediction, where a model is designed to predict the pitch/fret combination for each note of a musical score. This problem has been studied with path-finding algorithms [13] and minimax techniques [14]. More recently, Cheung et al. [15] used a deep learning model to generate fingering annotations, though not for guitar but violin. Those instruments nonetheless share some properties, and a study of fingering prediction on string instruments like the violin or the guitar, compared to other instruments, can be found in [16]. However, the task of using symbolic music to study occurrences of playing techniques is rarely studied. Xie and Li [17] propose to predict playing techniques as a tagging task on symbolic bamboo flute music but, to the best of our knowledge, no such work has been proposed for guitar music.

3. MODELING BENDS IN TABLATURE

To formulate bend prediction as a machine learning task, we adopt a representation for bent notes consisting of four different labels. We also need to pre-process our data to obtain bend-less scores, and musical features that could be used to train a machine learning model. Those considerations are presented hereafter.

3.1 Labeling

In the introduction, we presented the different types of bends that can be encountered in guitar tablatures. Based on this taxonomy, we define 4 labels that represent the motion and current state of the played string:

- $\varnothing$ — the string is not bent;
- $\uparrow$ — the string is bent, causing the pitch to go up;
- $\rightarrow$ — the string was bent previously and is plucked again in that state. The pitch is constant, but is not the one expected from the note's string/fret position;
- $\downarrow$ — the string was bent and is released, making the pitch go down accordingly.

We define those labels to circumvent the issue with *up & down* and *complex bends* that are not transcribed consistently. Labeling notes with the associated string movements therefore permits representing bends accurately, without loss of generality. Using these labels, all non-bent notes will be labeled $\varnothing$, *basic upward bends* are labeled $\uparrow$, *held bends* correspond to $\rightarrow$, and *reverse bends* are $\downarrow$. To fit with this representation, *up & down bends* are split into two notes of equal duration, the first one being labeled with $\uparrow$, and the second one with $\downarrow$. An example of such labeling on an actual guitar track is shown on top of Figure 1.

While bends can be of different amplitude, we do not include that information in our labeling, as we do not aim at predicting it in this work but only focus on predicting when bends occur. Similarly, we do not distinguish single notes from chords when predicting bends. This means that when the system predicts the presence of a bend in a chord, it does not specify on which string it occurs. The impact of this simplification is however limited, as bends rarely occur inside chords (12% of bend events are found in chords in our corpus).

3.2 Deriving a bend-less score simplification

Since our goal is to predict whether a note is played with a bend, we must start from a *simplified* tablature that does not have any bend information. Removing bend annotations from a tablature is however not a straightforward task since this technique affects the pitch of the performed note. We design the following procedure when translating bent notes from the fret/string space to the pitch space:

- If a note is labeled by $\varnothing$, its pitch is directly obtained from the string/fret combination;

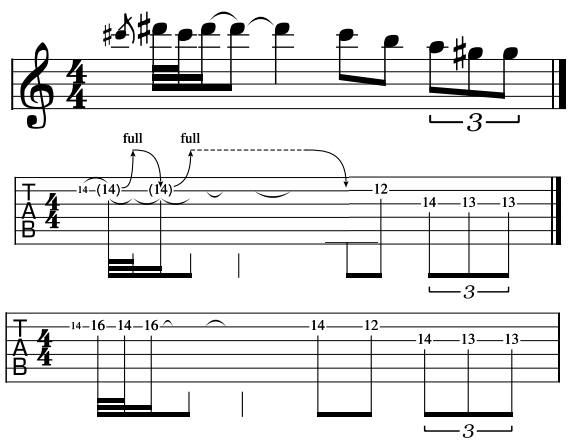


Figure 2: Excerpt of *Watermelon in Easter Hay*, Frank Zappa, as transcribed [18] by Steve Vai in standard notation (top). Below are two possible tablature representations of this excerpt with (middle), or without (bottom) bends.

Temporal	Duration
	Beat Strength
	Longer than previous
	Shorter than previous
	Same duration as previous
Pitch	Number of notes
	Pitch$^{(j)}$
	Pitch jump$^{(\mathrm{n}\pm k)}$
	Accidentals
	Pitch-class w.r.t scale root
Position	Fret$^{(\mathrm{n}\pm k)}$
	String$^{(\mathrm{n}\pm k)}$
	Fret jump $^{(\mathrm{n}\pm 2)}$
	String jump $^{(\mathrm{n}\pm 2)}$

Table 1: List of high-level features extracted from the note events. Let n denote the current note index, an exponent on a feature tells on which neighbors it is computed. $k \in \{1, 2\}$ because we discard any positional information related to the current note, and $j \in [\![\mathrm{n}-2, \mathrm{n}+2]\!]$. Other features are only computed on n.

- Otherwise, the pitch is the one of the bend arrival note. In particular:
  - if the label is $\uparrow$ or $\rightarrow$, the arrival pitch is the pitch of the string/fret combination, to which the bend amplitude is added;
  - if the label is $\downarrow$, the arrival pitch is the one corresponding to the string/fret position, because the string is released to its default state.

This approach is the one chosen by Steve Vai when transcribing Frank Zappa's melodies to standard notation, as we illustrate in Figure 2. The middle tablature shows how this excerpt is actually played (based on a live performance) and the bottom tablature illustrates how the same excerpt might be played without bends. While it is not an issue in this example, there is an uncertainty regarding where a bent note would be played on the fretboard, without a bend (keeping its destination pitch but losing the technique). A guitar player might indeed choose to play a note on a higher string, if remaining on the same string called for an uncomfortably large hand span. Because deciding arbitrarily of a hand position could introduce bias into our model, we choose not to include any string/fret information concerning the current note in the proposed features for our classification task, as explained hereafter.

3.3 Selected Features

To predict whether a note is bent, we propose an intermediate representation as a set of high-level features, presented in Table 1. Some descriptors focus on the event under scrutiny, while others provide short-term context information, both from the past and the future. Part of the features are derived from standard staff notation and convey *temporal* and pitch information, while others are related to *position* and the tablature space. If the studied event is a chord, the pitch, fret, and string values are averaged over all its constituting notes. While the average might seem an overly simple statistic, experiments with other functionals such as *min, max* or *standard deviation* did not improve the results. We have discarded any open strings for the fretboard features so that the average fret and string represents the actual position of the fretting hand. Apart from absolute/relative duration values, temporal features include the *beat strength*, which is a value between 0 and 1 suggesting how *strong* the beat of a note is. We obtain this value from the default implementation of the `music21` library [19]. These beat strength values have been designed for Western classical music, and therefore may be debatable for pop and rock music. However, they are mostly used here to represent onset times independently of the time signature, while grouping notes that share rhythmic properties.

In addition to the features on the current note, we extract a context of two past and two future note events, as preliminary experiments did not show any benefits of longer contexts. Additional boolean features are provided to recall if a neighboring note event is missing, when a note is preceded or followed by a whole rest for instance. When a note is missing, all corresponding features are set to 0. From this context, we compute the pitch jump between neighboring notes as well as the string and fret jumps when they are defined, *i.e.,* not with respect to the current note – because we do not know where the guitarist would play the note if they were to bend it. We expect these features to help our algorithm derive the hand position on the fretboard, which would be useful since bends are more likely to occur on certain spots of the fretboard, as will be shown subsection 4.2. Furthermore, we add information about the key signature through the number of accidentals (positive for sharps, negative for flats). From those accidentals, we derive the root note of the corresponding pentatonic minor scale (that scale encompassing much of guitar popular music [20]) and store the position of each note on this scale. For example, one sharp would make the root E, and an A would be numbered 5 since it is 5 semitones above E.

∅	↑	→	↓	Total
123 231	9627	1270	3314	137 442

Table 2: Number of notes per label in our dataset.

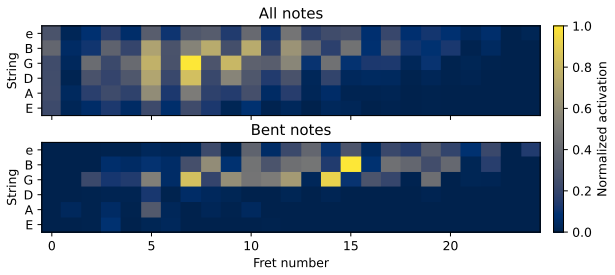


Figure 3: Normalized Heatmaps of all notes (Top) and bent notes (Bottom). The letters refer to the open string pitch in standard tuning with *e* being the high E string.

4. DATASET

4.1 Guitar Tablature Corpus

Our experiments are performed on the proprietary corpus *MySongBook* composed of 2247 guitar tablatures accurately transcribed by professional musicians in the `.gp` GuitarPro format. A subset of 932 tracks estimated as *lead guitar* – totaling more than 130 000 notes – was extracted by applying the classification technique from [21]. Our experiments focus on lead guitar parts, as they were felt to feature heavier use of playing techniques. In contrast with the whole corpus, which includes 2.5 % of bent notes, our lead guitar sub-corpus indeed contains 10% of bent notes, slightly mitigating the observed class imbalance.

Our work is implemented in *Python* and uses `music21` [19] and `scikit-learn` [22] libraries. To foster reproducibility, all our code is made publicly available (parsing of `.gp` files, extraction of features, training, and evaluation of bend classification models). We also release the complete set of features extracted on each note of our corpus, plus corresponding labels at:
`http://algomus.fr/code/`.

4.2 Statistical study

Table 2 reports the distributions of bend labels in our corpus. The distribution of bent notes on the fretboard, compared to all notes, is shown in Figure 3. We observe that most bends occur on the top 3 strings in the middle area of the fretboard. This observation differs from notes in general that are played on all strings, and especially on the two middle ones and around the 7$^{\text{th}}$ fret. While it is possible that the obtained heatmaps are biased by an over-representation of certain key signatures in the dataset – 43% of the tracks are in G major/E minor or C major/A minor – this bias should affect both heatmaps equally, so their mutual comparison is still possible. Because bent notes are found on both higher strings and higher frets than all notes, their pitch is similarly higher on average, as it can be observed in Figure 4a.

The distribution of *beat strength* values is shown in Figure 4b. Because most beats and sub-beats in a measure have a beat strength of 0.25 or below, no label is mostly played on strong beats.An interesting result is that ↑ and ↓ labels appear more often on stronger beats than ∅ and →. This apparent correlation of ↑ and ↓ labels with the meter might suggest a link between note expressiveness and accentuation in performance, which would need to be investigated further. In contrast, the → label is most often encountered on *weaker* beats. This observation can be linked to the fact that this technique is often used as a quick repetition of the previous note and will thus be played on the next offbeat, like in Figure 1.

The comparison of the duration of notes with or without bends (Figure 4c) confirms that ↑ and ↓ labels share some essential properties. Both labels have a proportionally higher tendency to be found on notes with longer duration, even though eighth note is the most common duration for all classes. This figure also confirms that ∅ and → classes share some context properties. Figure 4d shows a strong tendency of ↑ labels to appear on notes with longer duration than their predecessor. This further supports the hypothesis that bends could be used to emphasize significant notes in a lead guitar part. That result could also be related to the substantial physical effort required to bend a string on short duration notes. The accompanying code provides interactive computation of the distribution of the other features.

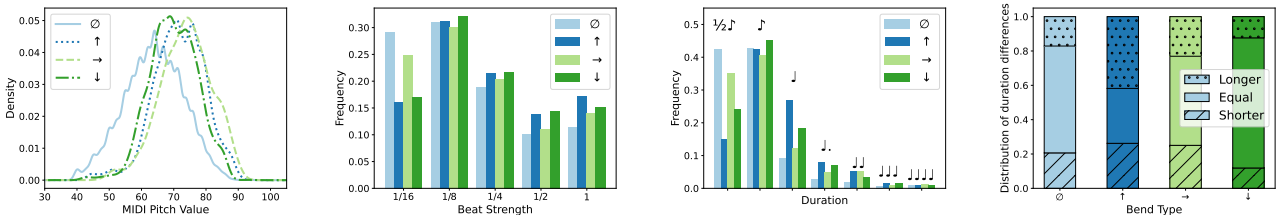
5. CLASSIFICATION RESULTS

A decision tree [23] was trained to predict the bend label of a note from its feature representation. We choose this high-level approach to facilitate the interpretation of the results as well as the analysis of the contribution of the features. In addition to the elaboration of a predictive model, conducting our experiments in an explainable AI framework allows us to improve our understanding of the use of bends in this repertoire. We hope that the use of light models enabled by highly expressive musical representations will also contribute to promoting low energy consumption approaches in machine learning for MIR.

5.1 Model performance

Our classifier is trained on 75% of the dataset, and evaluated on the remaining 25%. To avoid some leakage from the training set to the test set, we ensure the split does not separate notes from a same track. We also ensure that the class imbalance is similar in both sets. Duplicate feature vectors are removed track-wise to avoid overfitting due to repeated riffs/patterns. But, we acknowledge the fact that identical feature vectors can be found in different tracks and thus keep duplicates when found in different files.

The confusion matrix of Figure 5 shows the results of the multi-class classification on the joint prediction of all labels. Because the dataset is highly unbalanced, our model is naturally biased towards the ∅ label. However, it successfully identifies more than half of the ↑ and ↓ la-



(a) Pitch (b) Beat Strength (c) Duration (d) Duration diff. with previous

Figure 4: Distribution of four of the extracted features, normalized per bend class.

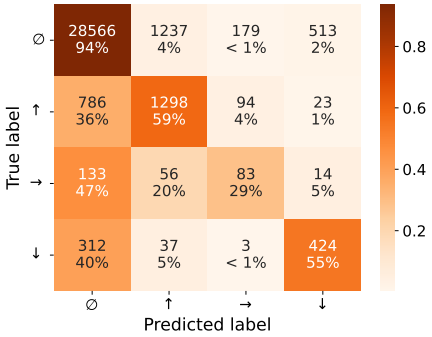


Figure 5: Confusion matrix obtained for classifying each note event to one of the bend class. This matrix was obtained on a split with average performance.

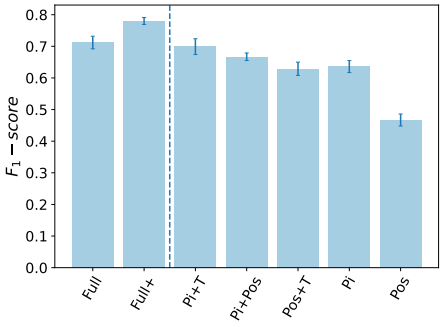


Figure 6: Average F_1-scores from 4 different train/test splits for the binary classification task. The leftmost part shows the performance of the decision tree trained on all features, with (Full+) or without (Full) SMOTE oversampling. The rightmost part corresponds to decision trees trained with a reduced set of features. *T* stands for temporal , *Pi* for pitch and *Pos* for position features.

bels. Samples labeled as → are often misidentified as ↑ but this result still shows, presumably, that the model captures the difference between ∅ and → labels. We tried applying SMOTE oversampling [24] to the training data and observed that it doubles the number of correctly identified → notes and increases the ratio of well-classified ↓ notes by approximately 10%. Nevertheless, ↑ notes *True Positives* (TP) ratio is about the same while the quantity of ↑ notes misidentified as → or ↓ increases. Similarly, TP ratio of ∅ notes drops by 5 p.p., so 2000 more notes are wrongly predicted as bent. Because we observed that bent notes are sparse in guitar tracks, we consider that *precision* is more important than *recall* and do not use any oversampling for the rest of our analysis.

5.2 Feature importance

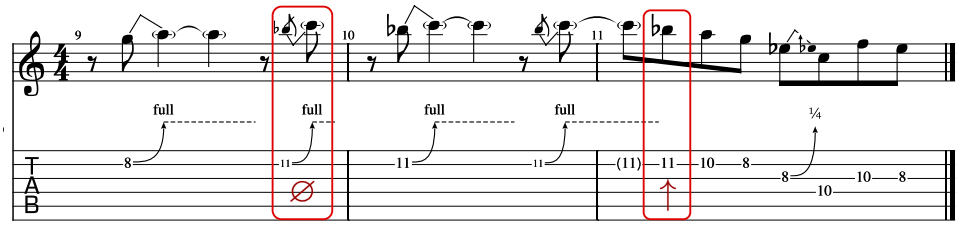
To assess the contribution of each feature, we conduct an *all bend* binary classification experiment where ↑, →, ↓ are merged into a single class, versus the ∅ class. Table 3 shows the importance of the eight most contributing features, computed using the random feature permutation technique introduced in [25] and monitoring its impact on the F_1 score of our model. Temporal and pitch features appear to have a higher impact on classification than position-related features, an observation confirmed by training the binary classifier on selected subsets of features. The results in Figure 6 confirm the dominant influence of pitch features. However, adding gesture and temporal information noticeably improve the results. This result suggests that, while fret context contributes to induce bent notes, a large part of the prediction can be done from the strict musical content as notated in musical scores.

6. PREDICTION ANALYSIS

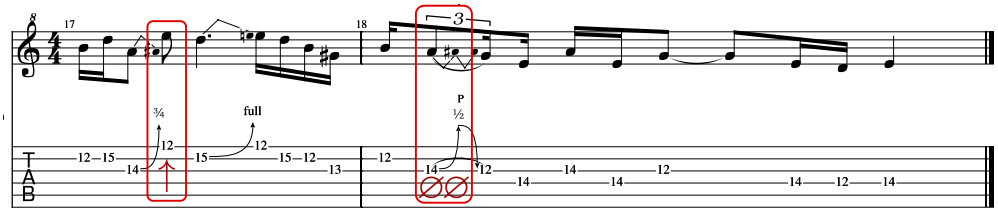
In addition to the quantitative results presented in the last section, we present a qualitative analysis of selected predictions. Figure 7a shows one bent note wrongly identified as ∅ and, conversely, one non-bent note identified as ↑ . Following the decision path provided by the decision tree, we can gain some insight on what feature differences have

Feature	Importance
Pitch	0.20
Pitch jump$^{(n+1)}$	0.17
Pitch jump$^{(n-1)}$	0.16
Duration	0.14
Same dur. as previous	0.07
Fret jump$^{(n-2)}$	0.07
String$^{(n+1)}$	0.05
Pitch$^{(n+1)}$	0.05

Table 3: Feature importance of the 8 most significant features for the decision tree. Standard deviation of any feature importance is never above 0.005.



(a) Excerpt from *Highway Star*, Deep Purple.



(b) Excerpt from *Jailbreak*, AC/DC.

Figure 7: Examples of predictions obtained with our Full Tree model on two different excerpts. Labels shown represent the predicted label for the current note. Only wrong predictions are shown for clarity. All other notes are labeled correctly.

caused those wrong predictions. Both notes actually have more than half their decision path in common and split on their Pitch jump$^{(n+2)}$ value, suggesting that the first discrepancy was due to future context. In particular, the second false prediction did not use any features related to past context. This might also explain this error because the pitch could be obtained by bending on the 10$^{\text{th}}$ fret by one semitone – an information that could be derived from future context – but continuity with the previous notes called for playing the note without bend on the 11$^{\text{th}}$ fret – an information that should have been derived from past context. Another observation is that, in spite of similar context, the second bent note was misidentified whereas the fourth bent note was not. While those two notes look very similar at first glance, the latter has a longer duration because it is tied to the following eighth note, which illustrates the importance of the *duration* featureAn analysis of the decision paths indeed shows a divergence from the second decision rule, based on that feature. This highlights the presumably strong influence of rhythm in the classification of the first four bent notes, which bypasses pitch features.

Figure 7b also shows a regular note wrongly tagged with a ↑ label. The decision path for this prediction does not consider any feature related to the next note. It does however use many features concerning the second next note, which was correctly classified as ∅ , most likely because of its lower duration. The lack of information about the current note's position was probably critical in that case. The second error on that tablature is an *up & down bend* that was not identified, probably because of the low duration of the involved notes. Nevertheless, this example suggests that our method to obtain a bend-less transcription from an up & down bend might be detrimental to the algorithm performance. Indeed, our procedure has an impact on *duration* and *pitch jump*$^{(\text{n}\pm1)}$ which are among the most useful features to our algorithm. We observe however that our algorithm predicted correctly six bend labels in the selected examples with a limited amount of false positives. These encouraging results suggest that our method could be used as a suggestion tool for the idiomatic use of bends.

7. CONCLUSION

In this paper, we proposed a model of guitar bends and discussed how these expressive playing techniques relate to both tablature and score content. Introducing a set of high-level features, we showed that a decision tree can successfully predict bend occurrences with satisfactory precision, in spite of the difficulty of the task due to the low proportion of bent notes in guitar music. In particular, the low performance on predicting → labels suggests that our modeling choices could be improved and that held bends might not be considered as an expressiveness technique but rather another way of playing regular notes. An advantage of our approach is the use of a lightweight and explainable algorithm, facilitating its use in an assisted-composition context. In future work, this approach could be extended to other guitar playing techniques, and might benefit from adding more context information like the chord being played over a bar, using rhythm guitar parts aligned with lead guitar. Because bends are arguably more easily performed with the ring finger and little finger than other fingers of the fretting-hand, combining our work with finger prediction technique [16] might also improve prediction performance. Finally, our modeling strategy could also be used to study the playing style of specific guitarists, and evaluate the potential of bends for automatic guitarist identification. Indeed, our approach supposes that bends can be explained by general musical features regardless of the artist. This is debatable since famous players can be identified by their solos (without considering audio nor any playing technique) [26]. It would be interesting to see if bends are artist-dependent and, if so, to develop a model that predicts bends in the style of a specific guitarist.

Acknowledgements. This work is made with the support of the French National Research Agency, in the framework of the project TABASCO (ANR-22-CE38-0001). The authors would like to thank Arobas Music for providing the dataset and colleagues from the Algomus team for their thorough proofreading and insightful comments.

8. REFERENCES

[1] D. R. Grimes, "String Theory - The Physics of String-Bending and Other Electric Guitar Techniques," *Public Library of Science One*, vol. 9, no. 7, Jul. 2014.

[2] P. J. Gomez, "Modern Guitar Techniques; a view of History, Convergence of Musical Traditions and Contemporary Works (A guide for composers and guitarists)," Ph.D. dissertation, UC San Diego, 2016.

[3] Q. Xi, R. M. Bittner, J. Pauwels, X. Ye, and J. P. Bello, "Guitarset: A Dataset for Guitar Transcription," in *Proc. of the 19th International Society for Music Information Retrieval Conference*, Paris, France, 2018.

[4] A. Wiggins and Y. Kim, "Guitar Tablature Estimation with a Convolutional Neural Network," in *Proc. of the 20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands, 2019.

[5] L. Reboursière, S. Dupont, O. Lähdeoja, C. Picard-Limpens, T. Drugman, and N. Riche, "Left and right-hand guitar playing techniques detection," *NIME*, 2012.

[6] S.-H. Chen, Y.-S. Lee, M.-C. Hsieh, and J.-C. Wang, "Playing Technique Classification Based on Deep Collaborative Learning of Variational Auto-Encoder and Gaussian Process," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*, Jul. 2018, pp. 1–6.

[7] Yu-Fen Huang, Jeng-I Liang, I-Chieh Wei, and Li Su, "Joint analysis of mode and playing technique in Guqin performance with machine learning," in *Proc. of the 21st International Society for Music Information Retrieval Conference*, Montréal, Canada, 2020.

[8] P. Sarmento, A. Kumar, C. J. Carr, Z. Zukowski, M. Barthet, and Y.-H. Yang, "DadaGP: A Dataset of Tokenized GuitarPro Songs for Sequence Models," in *Proc. of the 22nd Int. Society for Music Information Retrieval Conf.*, Online, 2021.

[9] P. Sarmento, A. Kumar, Y.-H. Chen, C. Carr, Z. Zukowski, and M. Barthet, "GTR-CTRL: Instrument and Genre Conditioning for Guitar-Focused Music Generation with Transformers," in *Artificial Intelligence in Music, Sound, Art and Design*, ser. Lecture Notes in Computer Science, C. Johnson, N. Rodríguez-Fernández, and S. M. Rebelo, Eds. Cham: Springer Nature Switzerland, 2023, pp. 260–275.

[10] Y.-H. Chen, Y.-H. Huang, W.-Y. Hsiao, and Y.-H. Yang, "Automatic Composition of Guitar Tabs by Transformers and Groove Modeling," in *Proc. of the 21st International Society for Music Information Retrieval Conference*, Montréal, Canada, 2020.

[11] M. McVicar, S. Fukayama, and M. Goto, "AutoLeadGuitar: Automatic generation of guitar solo phrases in the tablature space," in *2014 12th International Conference on Signal Processing (ICSP)*. Hangzhou, Zhejiang, China: IEEE, Oct. 2014, pp. 599–604.

[12] J. Ching and Y.-H. Yang, "Learning To Generate Piano Music With Sustain Pedals," in *Extended Abstracts for the Late-Breaking Demo Session of the 22nd Int. Society for Music Information Retrieval Conf.*, 2021.

[13] S. I. Sayegh, "Fingering for String Instruments with the Optimum Path Paradigm," *Computer Music Journal*, vol. 13, no. 3, pp. 76–84, 1989.

[14] G. Hori and S. Sagayama, "Minimax Viterbi algorithm for HMM-based Guitar fingering decision," in *Proc. of the 17th International Society for Music Information Retrieval Conference*, 2016.

[15] V. K. M. Cheung, H.-K. Kao, and L. Su, "Semi-supervised violin fingering generation using variational autoencoders," in *Proc. of the 22nd International Society for Music Information Retrieval Conference*, Online, 2021.

[16] G. Hori, "Three-Level Model for Fingering Decision of String Instruments," in *Proc. of the 15th International Symposium on CMMR*, Online, 2021.

[17] Y. Xie and R. Li, "Symbolic Music Playing Techniques Generation as a Tagging Problem," Oct. 2020, preprint.

[18] F. Zappa and S. Vai, *The Frank Zappa Guitar Book*. Hal Leonard Corporation, 2017.

[19] M. S. Cuthbert and C. Ariza, "music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data," in *Proc. of the 11th International Society for Music Information Retrieval Conference*, 2010.

[20] D. Temperley, *The Musical Language of Rock*. Oxford University Press, Jan. 2018.

[21] D. Régnier, N. Martin, and L. Bigo, "Identification of rhythm guitar sections in symbolic tablatures," in *Proc. of the 22nd International Society for Music Information Retrieval Conference*, Online, 2021.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.

[23] L. Breiman, *Classification and regression trees*. Routledge, 2017.

[24] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Oversampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002.

[25] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[26] O. Das, B. Kaneshiro, and T. Collins, "Analyzing and classifying guitarists from rock guitar solo tablature," in *Proceedings of the Sound and Music Computing Conference*, Limassol, Chypre, 2018.