Artificial intelligence assisted violin performance learning

Hannah Zhang[1], Nisha Talagala[2]
[1] BASIS Oro Valley, Tucson, Arizona
[2] AIClub, Pyxeda AI, Saratoga, California

SUMMARY

While learning the violin has many benefits, some people don't have access to teachers. Furthermore, music students typically have a week in between lessons, during which time they lack guidance – feedback that can greatly improve performance – on their practice. Our study explores an "artificial music tutor" for assisting violinists. We aimed at detecting major solfège errors in instrumental play. We hypothesized that artificial intelligence (AI) algorithms Random Forest, K Nearest Neighbor, and Multilayer Perceptron would identify errors. We further hypothesized that Random Forest would perform the best given its viability across applications requiring classification tasks. To build the AI, we gathered violin recordings played correctly and with errors, which we translated into a featurized dataset. We trained AI algorithms to distinguish between correct and incorrect recordings using two datasets. The first dataset contained one correct recording and three incorrect intonation, rhythm, and tempo recordings per piece. We achieved 100% accuracy with Random Forest in detecting recordings. The second dataset contained two correct recordings and three incorrect intonation, rhythm, and tempo recordings per piece. The highest accuracy was 71.42% for distinguishing between both incorrect tempo vs. intonation with Random Forest and incorrect rhythm vs. intonation with Multilayer Perceptron. Our findings support our hypothesis that Random Forest is generally the most accurate algorithm. However, Multilayer Perceptron also achieved the highest accuracy for the second dataset, so we concluded it is suitable for identifying performance errors. This AI will assist musicians in practicing without a teacher, showing players where they can improve.

INTRODUCTION

Music is one of the most engaging intellectual activities for people of all ages. Learning an instrument has many benefits, such as stimulating brain cells, improving functions like memory and abstract reasoning skills, and providing an outlet for creativity and emotions (1, 2). In a nationwide study of 1,000 teachers and 800 parents, 89% of teachers and 82% of parents rated music education highly as a source for greater student creativity (3). However, there are people that don't have the luxury of having a music teacher nearby or don't have the financial means to attend music lessons. According to the Grammy Music Education Coalition, 3.8 million preK-12 students in the United States do not have access to music education (3). Furthermore, learning an instrument requires daily practice and instruction from a teacher. Music students typically have a week in between lessons where they get feedback from a teacher; but during that week, they have no guidance whatsoever on how they are playing their instrument.

This study explores the possibility of an "artificial music tutor" for assisting a violinist during their practice time whenever a human tutor is not available. We aimed to detect major solfège errors, which are errors related to the reading and execution of a musical sequence independent from the instrument played. Examples include tone error, where notes are played incorrectly, and duration error, where notes and rests last longer or shorter than they should (4). The manifestation of these errors results in incorrect playing.

There have been previous efforts made in the digital music learning field (5). Tonara360 offers an app featuring a scoring system that measures how students perform and rates them accordingly, which is where artificial intelligence (AI) comes into play (6). However, the app does not go into the details of what errors the player has made. SmartMusic's Practice App facilitates feedback on students' individual performances through an algorithmic assessment of note pitch, rhythm, and duration via computer processing tools, but not using AI (7). AI is the human-like ability of machines to interpret data and act intelligently (8). We chose to use AI for this study because it mimics the learning process of a human, embodying the intellectuality of a teacher more accurately than computer processing tools, which typically do not make decisions or exhibit the ability to learn and distinguish patterns (9). AI's similarity to human ability would allow it to adapt to specific users and provide tailored feedback. We focused on machine learning (ML), a branch of AI that uses data and algorithms to learn and improve accuracy – much like a human (10). We applied ML algorithms – computer programs that carry out the processes described above – to analyze performance errors made by musicians.

In order to provide relevant critique for aspiring violinists, we developed our AI to detect errors in three possible areas of music performance: (i) intonation, (ii) rhythm, and (iii) tempo. With the aid of supervised machine learning algorithms, we viewed the detection accuracy of musical errors in a piece using data collected from our own tests. When the accuracy reached a sufficient level, we started planning the integration of this AI into an online application for musicians, with the aim to provide remote assistance and improve practice time.

Our goal in this study was to determine whether machine learning algorithms can detect intonation, rhythm, and tempo errors in music performance. In comparing three supervised learning algorithms, we will choose the one that achieves the highest accuracy. The supervised learning algorithms that were selected for the analysis work are Random Forest, K

Nearest Neighbor (KNN), and Multilayer Perceptron (MLP) – chosen for their robust history in a variety of supervised classification problems (11-13).

Random Forest, a machine learning algorithm used for both classification and regression, is centered around the concept of ensemble learning – a process of combining multiple classifiers to solve complex problems and improve the performance of a model. Random Forest contains numerous decision trees on various subsets of the given dataset, taking the average to improve the dataset's predictive accuracy (14). This algorithm has impressive adaptability, handling binary, categorical, and numerical features, and there is little pre-processing that needs to be done – the data does not need to be rescaled or transformed (15). For these reasons, we hypothesized that Random Forest would most accurately identify performance errors.

KNN is a supervised machine learning algorithm. It is easy to implement in its basic form yet performs complex classification tasks. It uses all of the data for training while classifying a new data point or instance. KNN is a non-parametric learning algorithm, which means it does not assume anything about the underlying data. This is a useful feature because most real-world data does not follow any theoretical assumptions (16). Because of this, we hypothesized that KNN would be able to identify performance errors. However, we hypothesized that Random Forest would perform more accurately due to its suitability for our targeted classification problem.

MLP relies on an underlying Neural Network to perform classification tasks. It is characterized by several layers of input nodes connected as a directed graph between the input and output layers. This algorithm can be applied to complex non-linear problems, and it also works well with large input data with relatively faster performance (17). While neural networks have been shown to outperform machine learning algorithms across many industry domains, we don't know how the individual neurons work together to arrive at the final output. They keep learning until they come up with the best set of features to obtain a satisfying performance, but in doing so, they scale variables into a series of numbers that once the learning stage is finished, the features become indistinguishable (18). Hence, for this study, we believed a traditional network would be better because the study required an understanding of the variables at play. We hypothesized that MLP would be able to identify performance errors; however, because of its ambiguity, we did not select MLP as the algorithm that would most accurately predict performance errors.

We hypothesized that these three AI algorithms would help identify performance errors made by musicians. We further hypothesized that Random Forest would perform more accurately than the others given its general viability across applications. Our metrics are the commonly used evaluation techniques for supervised learning, namely accuracy, which measures the ability of ML models to make correct predictions, and the confusion matrix, which shows an algorithm's sensitivity (proportion of positive cases predicted as positive) and specificity (proportion of negative cases predicted as negative), which are measures used to define the performance of an algorithm (19, 20). We used two datasets to conduct this study. Our original dataset contained a correct recording and three separate incorrect recordings of intonation, rhythm, and tempo for each excerpt from 13 different Romantic Period pieces. Our expanded dataset contained two correct recordings and three separate incorrect intonation, rhythm, and tempo recordings for each excerpt from the original dataset and 15 new Contemporary Period pieces. We included two correct versions in the expanded dataset to see how the ML models respond to variants of a correct performance and accommodate different playing styles of each user. We also decided to incorporate a new genre for the expanded dataset to ensure the ML models receive discriminative information from the training set to avoid bias and potential overfitting, which is when the model learns noise and cannot generalize on new, unseen data (21).

Our results supported our hypothesis that classification algorithms are able to predict music errors, with the Random Forest algorithm delivering the most accurate results for the original dataset, and MLP and Random Forest delivering the most accurate results for the second, expanded dataset. For the original dataset, we achieved 100% accuracy with Random Forest in detecting the different versions of each piece. For the expanded dataset, the highest accuracy we achieved was 71.42% in distinguishing between both incorrect tempo vs incorrect intonation with Random Forest and incorrect rhythm vs incorrect intonation with MLP.

RESULTS

We applied three machine learning algorithms – Random Forest, KNN, and MLP – to detect intonation, rhythm, and tempo errors in a musician's playing using two datasets, which were both recorded by a high school player with 9 years of experience. Our first dataset contained 13 pieces from the Romantic Period, and our second dataset contained 28 pieces – a combination of the original dataset and 15 new Contemporary pieces. Both datasets contained separate incorrect intonation, rhythm, and tempo recordings for each piece. Dataset 1 contained one correct recording for each piece, and Dataset 2 contained two correct recordings for each piece. Prior to testing, we converted the audio files into features and retrieved the numerical data for intonation, rhythm, and tempo, which was given to the algorithms (**Figure 1**).

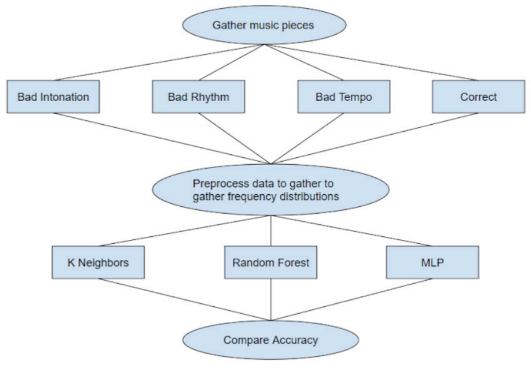

Figure 1: Schematic for Study Workflow. Flow chart of the experimental process. Describes the steps of the study, from gathering and preprocessing data to testing algorithms for musical error detection accuracy.

Experiment Results: Dataset 1

We first tested the ability of Random Forest, KNN, and MLP to detect errors in Dataset 1, which consisted of 13 Romantic Period songs, each with one correct recording and three separate incorrect intonation, rhythm, and tempo recordings. Random Forest achieved the highest accuracy, distinguishing between correct and incorrect recordings (along with their respective errors) with a value of 81.81% (**Table 1**). KNN followed with an accuracy of 45.45%. MLP had the lowest accuracy with a value of 36.36%. All accuracy values were calculated by measuring the number of recordings in the test set that the algorithm classified correctly out of the total test set. During this process, we generated a confusion matrix for each test. We also altered the hyperparameters for each algorithm, which changed the accuracy significantly in some cases. Hyperparameters are parameters whose values control the learning process of the model and determine the values of model parameters that an algorithm ends up learning (22). For both Dataset 1 and Dataset 2, we varied the same parameters for Random Forest, KNN, and MLP. For Random Forest, which utilizes a group of decision trees and combines their results to create the final prediction, the hyperparameter we used was number of trees, which determines the number of decision trees that are aggregated into the final result (23). The hyperparameter for KNN, which classifies a new data point based on how previously evaluated neighbor data points were classified, was k, which refers to the number of nearest neighbors to include in the classification process for a new data point (24). The hyperparameters we used for MLP – an artificial neural network that consists of neurons, or multiple layers of interconnected nodes – were maximum iterations, learning rate, and hidden layers (25). Maximum iterations are the maximum number of times a batch of data passes through the algorithm (26). Learning rate defines the pace at which an algorithm learns the values of a parameter estimate, which describe the contribution size of a predictor, and hidden layers are the layers in between the input and output layers where neurons perform computations on the data (27).

For Dataset 1, we increased the number of trees for Random Forest by an interval of five and found that the accuracy increased, leading us to deduce a positive correlation between number of trees and accuracy for our first dataset. After changing the value of k for KNN, we saw no patterns emerge despite starting with a low accuracy value. Hence, we decided this was not the right algorithm for our dataset. For MLP, no clear pattern of accuracy emerged from raising or lowering the maximum iterations. We varied the initial learning rate of the model – however this did not make a difference in accuracy. We also varied hidden layers from (100) to (100,100). The (100) notation represents a model with a single hidden layer of 100 neurons, while the (100, 100) notation represents two hidden layers of 100 neurons each.

Algorithm	Hyperparameters	Accuracy (%)
Random Forest	num_trees: 11; max_depth: null	90.9091
	num_trees: 15; max_depth: null	90.9091
	num_trees: 10; max_depth: null	81.8182
	num_trees: 20; max_depth: null	100.0000
	num_trees: 25; max_depth: null	100.0000
	num_trees: 30; max_depth: null	100.0000
KNN	k: 8	18.1818
	k: 3	27.2727
	k: 6	54.5455
	k: 4	27.2727
	k: 7	18.1818
	k: 5	45.4545
MLP	hidden_layers: 100 learning_rate_init: 0.01 max_iterations: 200	45.4545
	hidden_layers: (100,100) learning_rate_init: 0.01 max_iterations: 500	45.4545
	hidden_layers: (100,100) learning_rate_init: 0.01 max_iterations: 500	36.3636
	hidden_layers: (100,100) learning_rate_init: 0.01 max_iterations: 500	36.3636
	hidden_layers: 100 learning_rate_init: 0.001 max_iterations: 200	36.3636
	hidden_layers: (100,100) learning_rate_init: 0.01 max_iterations: 500	36.3636
	hidden_layers: 100 learning_rate_init: 0.0001 max_iterations: 200	36.3636

Table 1: Performance metrics for Dataset 1 of the three algorithms. Hyperparameters and accuracy values for Random Forest, KNN, and MLP distinguishing between correct recordings and incorrect intonation, rhythm, and tempo recordings, as tested on Dataset 1.

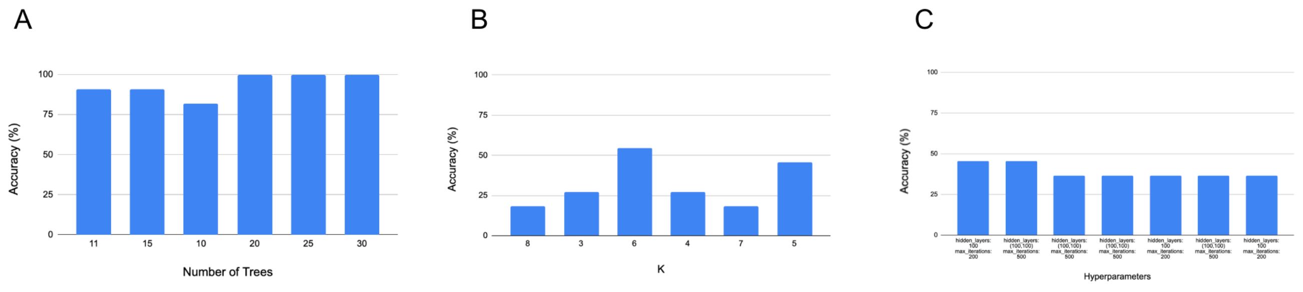

Figure 2: Algorithm accuracies for Random Forest, KNN, and MLP based on various hyperparameters for Dataset 1. A) Accuracy for Random Forest based on various numbers of trees for Dataset 1, which consists of excerpts from 13 Romantic Period violin pieces, for distinguishing between correct recordings and incorrect intonation, rhythm, and tempo recordings. **B)** Accuracy for KNN based on different values of k (number of nearest neighbors) for Dataset 1 for distinguishing between correct recordings and incorrect intonation, rhythm, and tempo recordings. **C)** Accuracy for MLP based on different values of hidden layers and maximum iterations for Dataset 1 for distinguishing between correct recordings and incorrect intonation, rhythm, and tempo recordings.

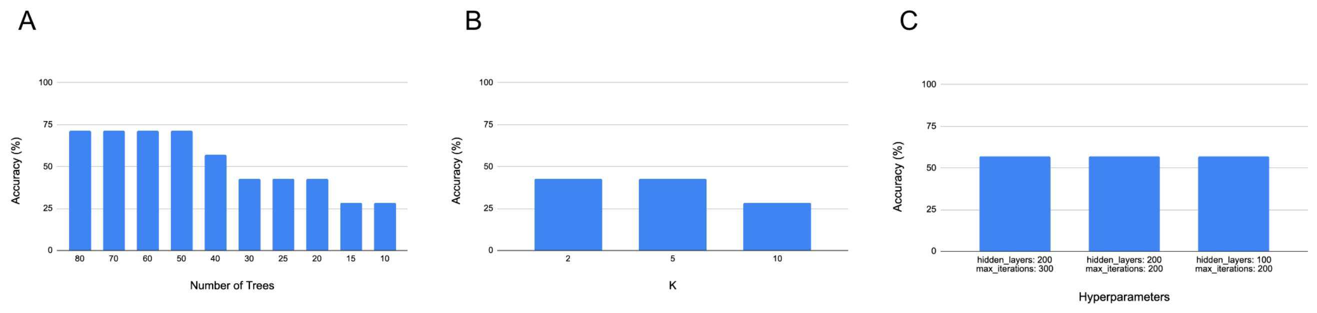

Figure 3: Algorithm accuracies for Random Forest, KNN, and MLP based on various hyperparameters for Experiment 1 of Dataset 2 (incorrect tempo vs incorrect intonation). A) Accuracy for Random Forest based on various numbers of trees for Dataset 2 (a combination of Dataset 1 and 13 new Contemporary piece excerpts), distinguishing between incorrect tempo recordings and incorrect intonation recordings. **B)** Accuracy for KNN based on various values of k for Dataset 2, distinguishing between incorrect tempo recordings and incorrect intonation recordings. **C)** Accuracy for MLP based on different values of hidden layers and maximum iterations for Dataset 2, distinguishing between incorrect tempo recordings and incorrect intonation recordings.

The (100, 100) notation represented a deeper network than the single hidden layer of 100 neurons, so we expected that incorporating more hidden layers would increase accuracy, as the model would be able to learn more complicated representations of the input data and catch complex patterns. However, we concluded that it did not affect the accuracy of MLP for Dataset 1. After varying these hyperparameters, we reached an accuracy of 100% with Random Forest and 54.54% with KNN (**Table 1, Figure 2A-B**). For MLP, we discovered that changing the hyperparameters did not lead to increased accuracy, remaining at 45.45% (**Table 1, Figure 2C**).

Overall, our results from Dataset 1 show that the AI is capable of detecting correct pieces from incorrect pieces in music performance. With an accuracy of 100% for the first dataset (to differentiate among the one correct and three incorrect versions for each piece), it seems that the AI functions with exceptional prediction accuracy.

Experiment Results: Dataset 2

For Dataset 2, we tested the ability of Random Forest, KNN, and MLP to detect between incorrect tempo versus incorrect intonation and incorrect rhythm versus incorrect intonation in our 28 songs dataset. Each song had two correct recordings and three separate incorrect intonation, rhythm, and tempo recordings. After reviewing the confusion matrices from Dataset 1 testing, we noticed that although we submitted equal numbers of incorrect intonation, rhythm, and tempo recordings to the algorithms, they were testing on fewer samples of intonation errors than samples of rhythm and tempo errors. We dedicated our Dataset 2 testing to see if this was because the algorithms were unable to differentiate intonation errors from rhythm and tempo errors. We decided to expand into the Contemporary genre for this dataset to ensure the models received a diverse training set to avoid overfitting. Additionally, we included two correct versions in Dataset 2 to see how the AI models respond to different versions of a correct performance and a user's individual playing style. Instead of taking the raw numerical values we obtained for each parameter, we took the average difference of the librosa parameters between the two correct recordings and each incorrect recording to feed to the classification algorithms. For example, librosa – a python package that we used to retrieve the numerical data for the intonation, rhythm, and tempo of our audio files – extracted a parameter called beats, however we didn't train the AI algorithms on the extracted feature for beats like we did in Dataset 1. We trained the algorithms on the average difference between the beats extracted for the two correct recordings and each incorrect recording for Dataset 2. We tested on the average difference of the librosa parameters because the algorithms would be able to compare the numerical intervals between the correct and incorrect versions rather than their raw numbers. We were curious to see how this would affect our accuracy for Dataset 2 since it served as a different comparison method for our models.

For the first experiment, we tested the three algorithms (i.e. Random Forest, KNN, and MLP) to differentiate between incorrect tempo and incorrect intonation (**Table 2**). For Random Forest, we found that increasing the number of trees led to an increase in accuracy. Since Random Forest combines predictions from multiple decision trees, with a larger number of trees, Random Forest can catch a greater variety of patterns, leading to better prediction accuracy. However, after 70 trees, there was no change in accuracy. For KNN, increasing the number of neighbors, or the value of k, led to either a stable or decreased accuracy. We concluded that KNN functions better with less neighbors for detecting between incorrect tempo and intonation for Dataset 2. For MLP, no patterns emerged while changing the hyperparameters. Random Forest had the greatest accuracy at 71.42% (**Figure 3A**). KNN had an accuracy of 42.85% (**Figure 3B**). MLP reached an accuracy of 57.14% (**Figure 3C**). For the second experiment, we tested how well the three algorithms differentiated between incorrect rhythm and incorrect intonation (**Table 3**). For Random Forest, increasing the number of trees led to an increase in accuracy, similar to the first experiment. For KNN, accuracy peaked at a value of 10 for k, functioning better with a greater number of neighbors than in Experiment 1 of Dataset 2. For MLP, hidden layers seemed to affect accuracy the most, leading to a decrease in accuracy as more neurons were added. Both Random Forest and KNN achieved a greatest accuracy of 42.85% (**Figure 4A-B**). We found that MLP achieved the highest accuracy overall with 71.42% (**Figure 4C**).

Overall, the results show that the AI is capable of

Algorithm	Hyperparameters	Accuracy (%)
Random Forest	num_trees: 80; max_depth: null	71.4286
	num_trees: 70; max_depth: null	71.4286
	num_trees: 60; max_depth: null	71.4286
	num_trees: 50; max_depth: null	71.4286
	num_trees: 40; max_depth: null	57.1429
	num_trees: 30; max_depth: null	42.8571
	num_trees: 25; max_depth: null	42.8571
	num_trees: 20; max_depth: null	42.8571
	num_trees: 15; max_depth: null	28.5714
	num_trees: 10; max_depth: null	28.5714
KNN	k: 2	42.8571
	k: 5	42.8571
	k: 10	28.5714
MLP	hidden_layers: 200 learning_rate_init: 0.001 max_iterations: 300	57.1429
	hidden_layers: 200 learning_rate_init: 0.001 max_iterations: 200	57.1429
	hidden_layers: 100 learning_rate_init: 0.001 max_iterations: 200	57.1429

Table 2: Performance metrics for Dataset 2, Experiment 1 of the three algorithms. Hyperparameters and accuracy values for Random Forest, KNN, and MLP distinguishing between incorrect tempo recordings and incorrect intonation recordings for Experiment 1 of Dataset 2.

Algorithm	Hyperparameters	Accuracy (%)
Random Forest	num_trees: 80; max_depth: null	71.4286
	num_trees: 70; max_depth: null	71.4286
	num_trees: 60; max_depth: null	71.4286
KNN	k: 2	28.5714
	k: 4	28.5714
	k: 5	28.5714
	k: 10	42.8571
	k: 20	28.5714
MLP	hidden_layers: 100 learning_rate_init: 0.0001 max_iterations: 300	71.4286
	hidden_layers: 100 learning_rate_init: 0.001 max_iterations: 300	71.4286
	hidden_layers: 100 learning_rate_init: 0.001 max_iterations: 250	71.4286
	hidden_layers: 150 learning_rate_init: 0.001 max_iterations: 200	57.1429
	hidden_layers: 100 learning_rate_init: 0.001 max_iterations: 200	71.4286

Table 3: Performance metrics for Dataset 2, Experiment 2 of the three algorithms. Hyperparameters and accuracy values for Random Forest, KNN, and MLP distinguishing between incorrect rhythm recordings and incorrect intonation recordings for Experiment 2 of Dataset 2.

categorizing different errors in music performance. With an accuracy of 71.42% for both experiments of Dataset 2 (to differentiate between incorrect tempo versus incorrect intonation and between incorrect rhythm versus incorrect intonation), it seems that the AI is able to differentiate intonation errors from rhythm and tempo errors with adequate accuracy.

DISCUSSION

For this study, we focused on building an "artificial music tutor" for assisting a violin performer when a human tutor is not available. Our findings supported our hypothesis that all three AI algorithms would be able to identify performance errors made by musicians, with Random Forest most accurately predicting these errors for Dataset 1 and Experiment 1 of Dataset 2. For Experiment 2 of Dataset 2, MLP performed the best among the three algorithms.

For both experiments of Dataset 2, our highest accuracy did not reach the caliber of Dataset 1. After examining the second dataset, we noticed this was due to a recurring problem the algorithms had with detecting correct recordings from incorrect recordings versus incorrect recordings from incorrect recordings. When compared to a correct recording, the AI had a standard to compare the incorrect recording to; however, this was not the case for comparing two incorrect recordings. This would make it harder to acquire a higher accuracy when comparing two incorrect recordings. In the future, we may try adding more musical recordings to combat this, giving the algorithms more data to analyze. Additionally, because we took the average difference between the librosa parameters for the two correct versions and each incorrect version of the recordings in Dataset 2, the AI algorithms were testing on the numerical intervals between the different versions rather than raw numbers of each version. However, we cannot be certain that the difference in preprocessing led to lower accuracy numbers for Dataset 2. In future studies, we plan to test this by taking the average difference of the librosa hyperparameters for Dataset 1 and comparing our algorithm results for these tests to our original Dataset 1 tests. For Dataset 2, despite submitting equal numbers of incorrect intonation, rhythm, and tempo recordings, the algorithms were not testing an equal number of rhythm and tempo errors compared to intonation errors, with markedly fewer samples of intonation errors. After Dataset 2 experimentation to see if the cause lay in the algorithms not being able to differentiate intonation errors from the other errors, we realized that this was not the issue. After further analysis, we determined that the AIs randomly generated a skewed matrix. Thus, we came to the conclusion that the second dataset would benefit from an increased number of samples to improve the accuracy of the AI with regard to intonation.

It is also important to note that the recordings used for Dataset 1 were all created from pieces composed during the Romantic Era, ranging from the 1800s to 1900s. Originating from Europe, the composers hailed from countries like Austria, Germany, France, and Russia. However, due to utilizing only Romantic Era pieces in Dataset 1, a conceptual issue arose in our study – using similar data when experimenting with AI algorithms often results in overtraining the model to the point that it cannot generalize on new data, leading to worse predictive performance than when using a diverse dataset. For example, if we were to test our model on a piece from the Contemporary Period, it would most likely perform poorly as it has only been exposed to the Romantic Period music from Dataset 1. Keeping this in mind, we added contemporary pieces to our second dataset. We plan to continue targeting

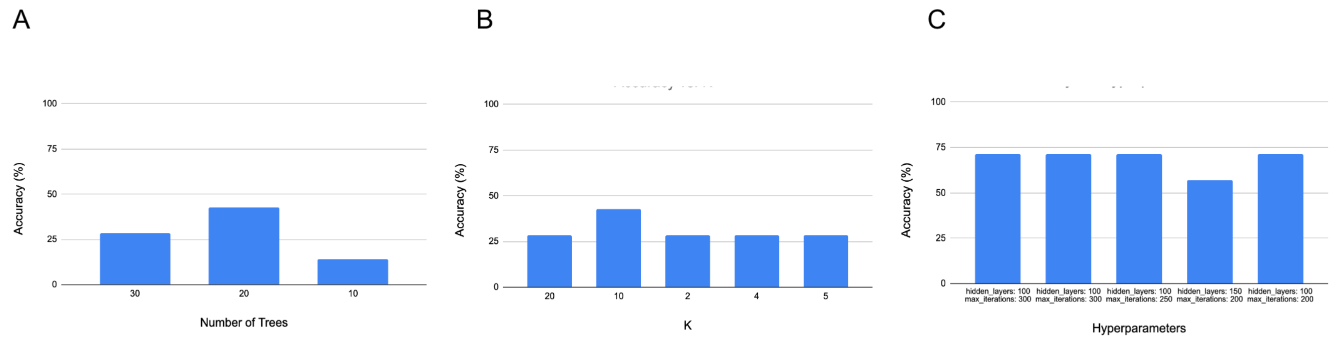

Figure 4: Algorithm accuracies for Random Forest, KNN, and MLP based on various hyperparameters for Experiment 2 of Dataset 2 (incorrect rhythm vs incorrect intonation). A) Accuracy for Random Forest based on various numbers of trees for Dataset 2 (a combination of Dataset 1 and 13 new Contemporary piece excerpts), distinguishing between incorrect rhythm recordings and incorrect intonation recordings. **B)** Accuracy for KNN based on various values of k for Dataset 2, distinguishing between incorrect rhythm recordings and incorrect intonation recordings. **C)** Accuracy for MLP based on different values of hidden layers and maximum iterations for Dataset 2, distinguishing between incorrect rhythm recordings and incorrect intonation recordings.

other genres of music in further studies. However, different genres have different characteristics and uses of intonation, rhythm, and tempo. For example, targeting the genre of contemporary music, where composers are abandoning tonal, structural, and rhythmical conventions, may prove difficult given its non-uniformity. This can be seen as a possible factor in our experiment results for Dataset 2, where our lower accuracy may be partially attributed to the analysis of unfamiliar, unpredictable types of error in contemporary music.

For future work, we hope to target other major errors that musicians make when practicing and performing, broadening the scope of the "artificial music tutor" and expanding the ways it can help the user. The current AI algorithm can only detect one type of error at a time in a recording. We plan to expand the AI algorithm to detect several different types of error concurrently in one piece. We will gather data by recording music pieces with multiple types of errors, such as a recording with both incorrect intonation and incorrect tempo and train the AI to detect these errors. However, when targeting songs with multiple errors, we would have to gather large amounts of data, which poses a challenge to carrying out the study with our current resources. In the future, we also plan to gather recordings of multiple musicians playing the same piece correctly and incorrectly so the AI can analyze variation in different musicians' playing. This would help the AI improve its ability to detect correct and incorrect versions of each piece. Still, this would require a larger dataset along with more AI training and testing, as we would most likely start out with lower accuracy numbers.

There are other errors in musical performance that we plan to address with our AI as well. One such example is dynamic errors, where the performer may ignore dynamic marks such as crescendo or piano forte. Besides Solfège errors, sound quality errors may manifest as non-musical sound (background noise error) or as a series of very short sounds indistinguishably separated (excess vibrato error) when a sound is incorrectly emitted. Another common situation is a lack of experience of a performer that leads them to an abrupt attack or decay of a note (incorrect attack/decay error). Lastly, some musical phenomena may be out of sequence (sequencing error) if the performer does not adequately read or follow repeat signs (such as da capo or dal segno) or breathing signs (for wind instruments). Similar to targeting songs with multiple errors, focusing on new errors would require large amounts of data, thus elongating the preparation process preceding the stage of AI training and testing.

MATERIALS AND METHODS

We gathered the violin recordings and preprocessed the data to gather frequency distributions. We tested our hypothesis that Random Forest would perform better than the other algorithms, analyzing the accuracy of the three classification algorithms – KNN, Random Forest, and MLP. The language we used for this study was Python and the packages were from Scikit Learn.

Datasets

For Dataset 1, we searched for a specific genre of music to concentrate the data to one time period and begin the early stages of testing. We chose the Romantic Period, and violin pieces from that time were added to the dataset. The player – a high schooler with nine years of experience – recorded four versions of an excerpt from each piece: incorrect intonation, incorrect rhythm, incorrect tempo, and fully correct. We used a total of 42 MP4 files across 13 songs. Dataset 2 contained a combination of Dataset 1 and 15 new Contemporary pieces, totaling 28 pieces. We defined Contemporary music as any piece written between the mid-1900s and modern day. The same player recorded five versions of an excerpt from each Contemporary piece, with two correct versions and three incorrect versions (one each for incorrect intonation, incorrect rhythm, and incorrect tempo). The player also recorded an additional correct version for all pieces in Dataset 1 that were added to Dataset 2. Using two correct versions for Dataset 2 would allow us to gauge more accurately how the AI algorithms respond to different versions of a correct performance, as players have different musical styles. In both datasets, each incorrect version had at most one type of error. These excerpts were recorded as MP4s using an audio recorder and later converted into numerical data and input into a CSV file. All of the recordings were done in a controlled environment with the same player, violin, room, and recording device.

Data Preprocessing

Using Python code in Google Colab, we converted raw MP4 files into numbers to be analyzed by the AI algorithms. We began by importing librosa, a python package for music and audio analysis (28). Librosa is used when working with audio data such as in music generation. It provides the building blocks necessary to create the music information retrieval systems. By using librosa, we were able to seamlessly convert the audio files into features and thus retrieve the numerical data for the intonation, rhythm, tempo, and correct versions of each song. The data was then turned into a CSV file that could be given to the AI algorithms to analyze. For the first dataset, we tested on the raw numerical values we obtained for each parameter. However, in the CSV files for the second dataset, we took the average difference of the librosa parameters for the two correct versions and each incorrect version of a music piece to give to the classification algorithms for testing.

AI Algorithm Application

With the newly acquired numerical values, we used Pyxeda Navigator to build the AI algorithms. Pyxeda Navigator is a web-based front end to a series of well-known machine learning and deep learning software packages (Scikit Learn, TensorFlow, Google Cloud Platform, and Amazon Sagemaker) (29). In our case, we invoked the algorithms from Scikit Learn. We input the CSV file and then set the prediction value to "label," which allowed the pieces to be differentiated into the three errors and the correct version. Following this, we trained the AI with the datasets using three algorithms – MLP, KNN, and Random Forest. For both datasets, the hyperparameters we used for MLP were maximum iterations, learning rate, and hidden layers. For Random Forest, we used number of trees. For KNN, the hyperparameter was k.

Received: May 26, 2022
Accepted: April 17, 2023
Published: August 30, 2023

REFERENCES

1. Miendlarzewska, Ewa A. and Trost, Wiebke J. "How musical training affects cognitive development: rhythm, reward and other modulating variables." *Front Neurosci.*, vol. 7, no. 279, Jan. 2014, https://doi.org/10.3389/fnins.2013.00279.
2. Ritter, Simone M. and Ferguson, Sam. "Happy creativity: Listening to happy music facilitates divergent thinking." *PLOS ONE*, vol. 12, no. 9, Sep. 2017, https://doi.org/10.1371/journal.pone.0182210.
3. Strauss, Valerie. "Perspective | Here's what's missing in music education: Cultural and social relevance." *The Washington Post.* www.washingtonpost.com/education/2019/07/19/heres-whats-missing-music-education-cultural-social-relevance/. Accessed 28 Apr. 2022.
4. Rangel, Nahum et al. "Deep symbolic processing of human-performed musical sequences." *Journal of Intelligent and Fuzzy Systems*, vol. 43, no. 6, Dec. 2021, https://doi.org/10.3233/JIFS-219261.
5. Holland, Simon. "Artificial Intelligence in Music Education: a critical review." *Miranda, E. (ed.) Readings in Music and Artificial Intelligence, Contemporary Music Studies*, vol. 20, pp. 239–274, Sep. 2000, https://doi.org/10.4324/9780203059746.
6. "Music Education Happens Here." *Tonara.* www.tonara.com. Accessed 28 Apr. 2022.
7. "Music Learning Software for Educators & Students." *MakeMusic*. www.smartmusic.com. Accessed 28 Apr. 2022.
8. Marr, Bernard. "5 Reasons Why Artificial Intelligence Really Is Going to Change Our World." *Forbes Magazine*. www.forbes.com/sites/bernardmarr/2020/05/08/5-reasons-why-artificial-intelligence-really-is-going-to-change-our-world/?sh=1076a32c78b6. Accessed 28 Dec. 2022.
9. Ongsulee, Pariwat. "Artificial intelligence, machine learning and deep learning." *2017 15th International Conference on ICT and Knowledge Engineering*, pp. 1-6, Nov. 2017, https://doi.org/10.1109/ICTKE.2017.8259629.
10. Zhou, Zhi-Hua. "Machine Learning." *Springer, Singapore*, 2022, pp. 1-24.
11. Gardner, M.W and Dorling, S.R. "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences." *Atmospheric Environment*, vol. 32, no.14-15, Aug. 1998, https://doi.org/10.1016/S1352-2310(97)00447-0.
12. Taunk, Kashvi et al. "A Brief Review of Nearest Neighbor Algorithm for Learning and Classification." *2019 International Conference on Intelligent Computing and Control Systems*, pp. 1255-1260, May 2019, https://doi.org/10.1109/ICCS45141.2019.9065747.
13. Pretorius, Arnu et al. "A meta-analysis of research in random forests for classification." *2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference,* pp. 1-6, Nov. 2016, https://doi.org/10.1109/RoboMech.2016.7813171.
14. Xu Gaowei et al. "Bearing Fault Diagnosis Method Based on Deep Convolutional Neural Network and Random Forest Ensemble Learning." *Sensors*, vol. 19, no. 5, Mar. 2019, https://doi.org/10.3390/s19051088.
15. Kho, Julia. "Why Random Forest Is My Favorite Machine Learning Model." *Medium*, Towards Data Science. towardsdatascience.com/why-random-forest-is-my-favorite-machine-learning-model-b97651fa3706. Accessed 28 Dec. 2022.
16. Goldberger, Jacob, et al. "Neighbourhood components analysis." *Neural Information Processing Systems,* pp. 513–520, Dec. 2004.
17. Jain, A.K. et al. "Artificial neural networks: a tutorial." *Computer*, vol. 29, no. 3, pp. 31-44, Mar. 1996, https://doi.org/10.1109/2.485891.
18. Buhrmester Vanessa et al. "Analysis of Explainers of Black Box Deep Neural Networks for Computer Vision: A Survey." *Machine Learning and Knowledge Extraction*, vol. 3, no. 4, Dec. 2021, https://doi.org/10.3390/make3040048.
19. Grandini, Margherita et al. "Metrics for Multi-Class Classification: an Overview." *arXiv,* Aug. 2020, https://doi.org/10.48550/ar756.Xiv.2008.05.
20. Navin, Maria et al. "Performance analysis of text classification algorithms using confusion matrix." *Int. J. Eng. Tech. Res.*, vol. 6, no. 4, Dec. 2016.
21. Bejani, Mahdi and Ghatee, Mehdi. "A systematic review on overfitting control in shallow and deep neural networks." *Artificial Intelligence Review*, pp.1-48, https://doi.org/10.1007/s10462-021-09975-1.

22. Agrawal, Tanay. "Hyperparameter optimization in machine learning: make your machine learning and deep learning models more efficient." *Apress*, 2021.
23. Zhou, Xiaoyi et al. "Accident prediction accuracy assessment for highway-rail grade crossings using random forest algorithm compared with decision tree." *Reliability Engineering & System Safety*, vol. 200, Aug. 2020, https://doi.org/10.1016/j.ress.2020.106931.
24. Taunk, Kashvi et al. "A brief review of nearest neighbor algorithm for learning and classification." *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, May 2019, https://doi.org/10.1109/ICCS45141.2019.9065747.
25. Zou, Jinming et al. "Overview of artificial neural networks." *Artificial neural networks: methods and applications*, vol. 458, pp. 14-22, 2008, https://doi.org/10.1007/978-1-60327-101-1_2.
26. Haykin Simon. "Neural networks: A Comprehensive Foundation." *Prentice Hall PTR,* 1998.
27. Goodfellow Ian et al. "Deep Learning." *MIT Press,* 2016.
28. "librosa." *Librosa*. librosa.org/doc/latest/index.html. Accessed 12 May. 2022.
29. "Navigator." *Pyxeda*. navigator.pyxeda.ai/login. Accessed 28 Apr. 2022.