Strumming Surveyal: Statistical Analysis of Guitar Performance via EMG Data

Alex Dyer and Simon Nunayon

***Abstract*—Guitar play involves intricate activation of the flexor digitorum profundus muscle to play desired strings in the intended style. Such intricacies elude superficial analysis. More advanced statistical techniques are then required to understand and predict player actions. This paper aims to employ various statistical methods to process and analyze guitar play. Specifically, the authors aim to employ PCA and K-means clustering to guitar string data and LDA to guitar style data. The team shall assess EMG data collected using the ES53 lab kit. Performance evaluation focuses on labeling accuracy and the predictive capacity of the systems. Further developments of this model are crucial to understanding guitar engagement and developing tools that can increase the accessibility and efficacy of guitar play.**

***Index Terms*—Guitar, Flexor Digitorum Profundus, PCA, LDA, K-means Cluster**

I. Introduction

ONE of the greatest challenges a recording artist faces is replicating a studio performance to a live audience. For guitar players, this is often achieved using a plethora of footswitches or changing playing style - using their fingers or a plectrum. In this project, we aimed to explore whether EMG data from the forearm could be used to determine a change in playing style, which could then be used to modify the sound in place of footswitches.

Fig. 1: Flexor Digitorum Profundus Involvement in Guitar Play

The flexor digitorum profundus (FDP) is the muscle in the forearm primarily responsible for the movement of the metacarpophalangeal and distal interphalangeal joints of the index, middle, ring, and little fingers. Such actions are crucial to the playing of stringed instruments. Our experimental set-up aims to monitor the electrical activity of this muscle to determine which string on a guitar is being played (e.g., A, D, E, etc.) and how the instrument is being played (by stringing with one's fingers or a pick). Therefore, the electrodes of the ES 53 kit are placed linearly on the anterior portion of the forearm. All of the data collected thus far were done in a continuous span of time to ensure controlled electrode placement. For each string of interest, 45 seconds of recording would begin, and the subject would play a given string repeatedly with approximately 4 seconds between each repetition. Following a break of around 3 minutes to mitigate the impact of fatigue, the subject would begin playing the next string. The same guitar was used throughout the study.

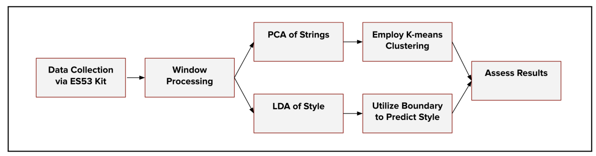

Fig. 2: Experimental Design Flow

4 different types of samples were collected:

1) Each finger playing a note on its respective string (1st finger 1st string etc.)
2) A plectrum being manipulated with finger and thumb to strike each string
3) All fingers being used to strum across all 4 strings
4) A plectrum being used to strum across all 4 stings

II. Data Filtering

First, we collated our data to give up 32 samples of each string played both with a pick and with fingers, then removed our recording signifiers. Two datasets were created: one with all fingerstyle notes and one for all notes played with a plectrum. These data were then filtered at 15 Hz using a 2nd Order Low-pass Butterworth filter. Next, we collated the samples of notes played on each string to form 8 datasets. These were also filtered at 15 Hz.

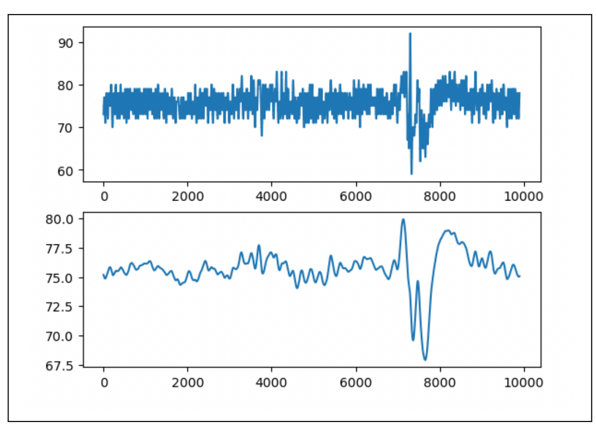

Fig. 3: Comparing Filtered Picked Note to Unfiltered Pick Note

As shown, the general shape of the signal is preserved while eliminating a large proportion of noise incurred by the ES53 sensor.

III. Window Processing

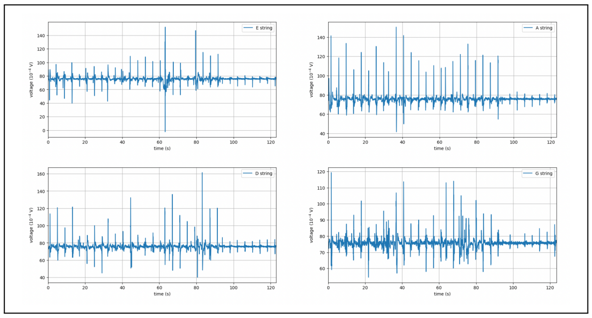

Fig. 4: EMG Signals for the Strings

Windowing the data after subtracting the mean provides some insight into the contraction pattern of each finger playing each string. Next is seeing if any significant trends can be spotted in the frequency domain.

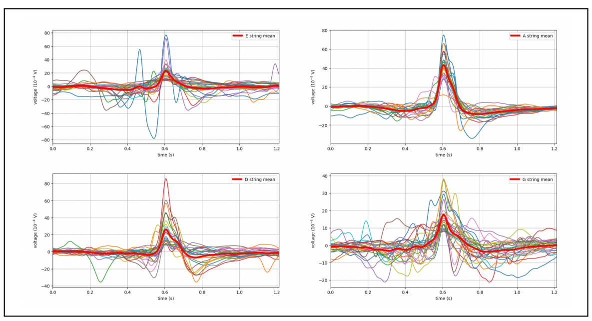

Fig. 5: Windowed EMG Data of the Strings

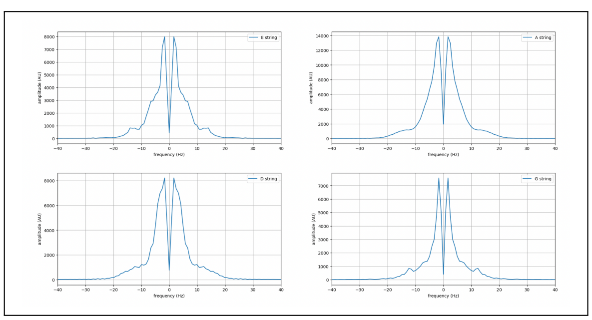

Fig. 6: FFT of Windowed EMG Data

IV. Principal Component Analysis

Principal component analysis (PCA) is a dimensionality reduction technique that enables the many-thousand index windows to be represented in a lower coordinate space. Such compression aims to maximize the variance encoded by each constituent coordinate. PCA offers a method to do so via a covariance matrix and then a change of basis.

With the windows isolated from the previous part, the authors concertante them into a single matrix. Calculating the

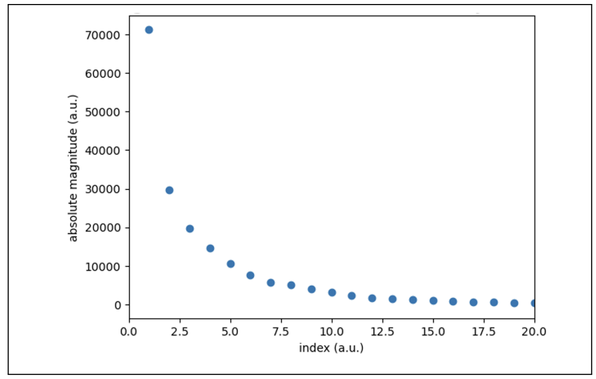

Fig. 7: Eigenvalues of Covariance Matrix

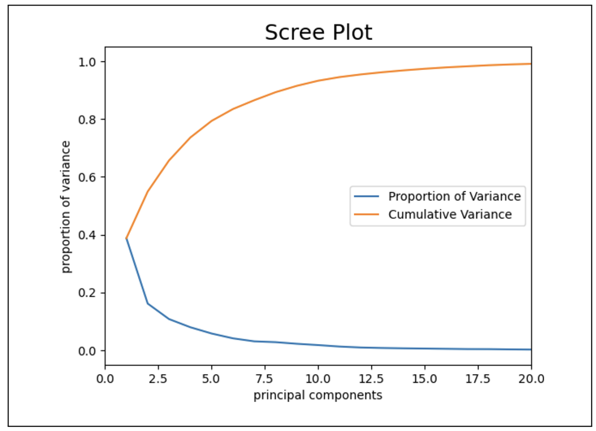

Fig. 8: PCA Scree Plot

covariance of this matrix will yield an n x n matrix such that n is equal to the number of indices in each window. To calculate the directions along which the variance across the dataset is maximized, the authors then calculate the eigenvalues and corresponding eigenvectors of the covariance matrix. The relative magnitude of the eigenvalues represents the amount of variance encoded by each eigenvector. The Figure "Eigenvalues of Covariance Matrix" illustrates the diminishing amount of variance captured by each eigenvalue/eigenvector pair, while the Scree Plot demonstrates this idea more explicitly. Projecting the windowed data onto the first two or three eigenvectors enables dimensionality reduction. Each eigenvector is then a principal component of the data. Based on the sample of data employed, the team notes that the first two principal components captured 61.58% of the data's variance, whereas the first three captured 73.17% of the variance. From these values and the Scree Plot, it is evident that the value of each additional component has diminishing marginal returns. Since both these reductions represent significantly more than half of the variation, the authors are confident to proceed with the string data analysis.

Projecting the data onto the first two and three principal components engenders a dimensionality reduction that enables the data to be represented visually. These representations can be seen in the scatterplot figures. Based on the results, one can observe distinct clusters of the data. These clusters are more distinct for the data represented in three dimensions.

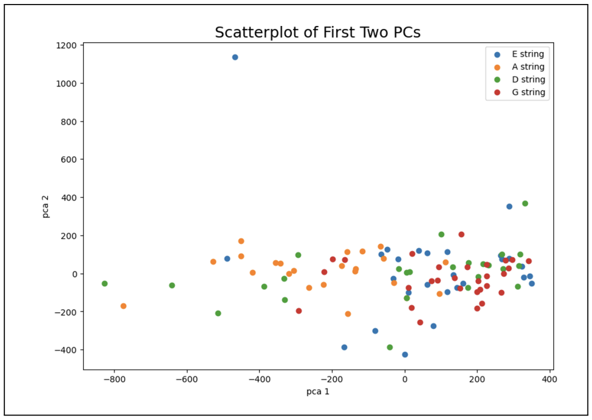

Fig. 9: PCA with 2 Components Scatterplot

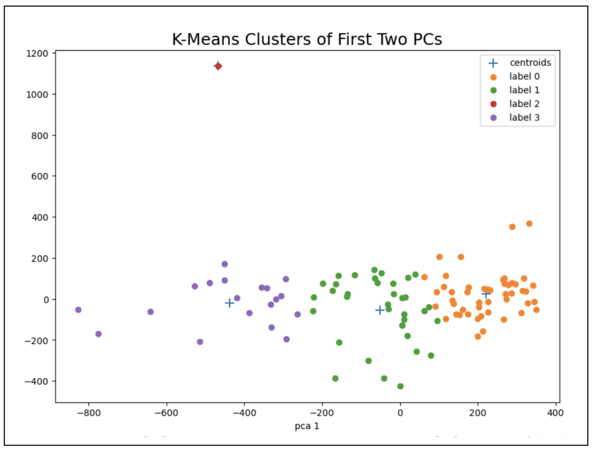

Fig. 10: K-means Clustering of 2D Data

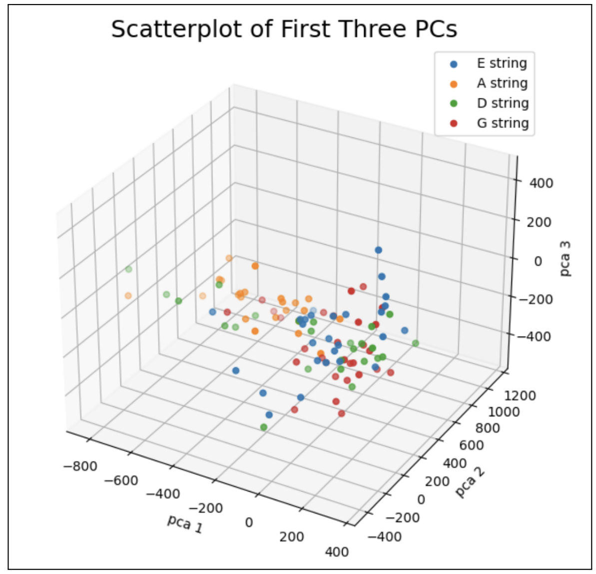

Fig. 11: PCA with 3 Components Scatterplot

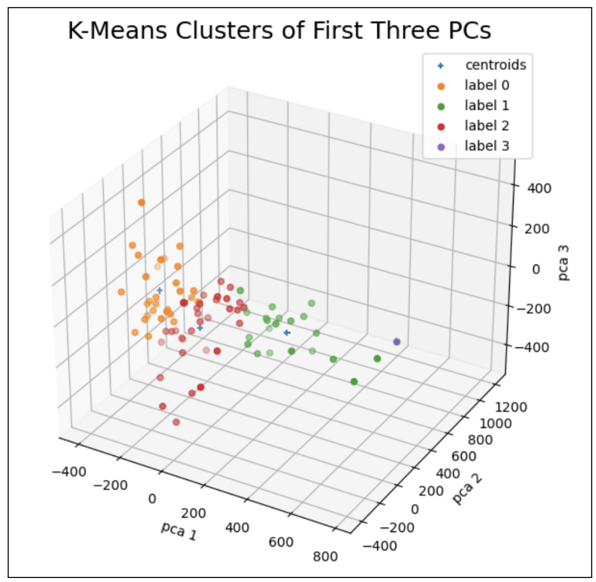

Fig. 12: K-means Clustering of 3D Data

This result makes sense, given the greater amount of variance preserved. Although the data exhibits distinguishable clusters, these clusters have a significant amount of overlap with each other. This overlap makes isolation and, thus, prediction of data difficult. To evaluate the extent of this overlap, the team will compare the labeled clusters with those generated from k-means clustering.

K-means clustering is an unsupervised machine learning method that aims to identify n clusters associated with an input number of n centroids. It does so by moving the centroids to minimize the distance that each closest point is to the centroid. The algorithm runs until the locations of the centroids stabilize. Each data point then becomes associated with a cluster. The team can assign each generated group to a string by employing the known classifications. After running the model, the authors note mixed results. As expected, the clusters provide data labels that ensure homogenous groups, which differs significantly from the overlap present in the PCA scatterplots. The results of the K-means clustering algorithm demonstrate a high susceptibility to outliers, as seen in how for both scatterplots, one of the clusters becomes assigned thereto. The three-dimensional data appears to have increased fidelity compared to the two-dimensional data.

Confusion matrices enable more rigorous analysis of the results. The matrices provide a pictorial representation of the correlation between the assigned labels and the known classes. Ideally, there should be a 1-to-1 correspondence that manifests as a high-valued diagonal. Although the confusion matrix each has a 1-to-1 correspondence between the rows and columns with the highest magnitude, there are significant off-axis values. As a result, the centroids do a poor job of isolating the clusters in alignment with the classes provided. In addition to the limited separability of the string data, another reason for this result is the unsupervised nature of the algorithm. From these matrices, one can tell that the three-dimensional data performed only slightly better. Although increased data cleaning could improve the results, a supervised separation method will likely provide improved results. On the topic of data cleaning, the limited precision of the electrode likely limits the results since it cannot detect the nuances behind fine motor movements. One interesting result is the analysis

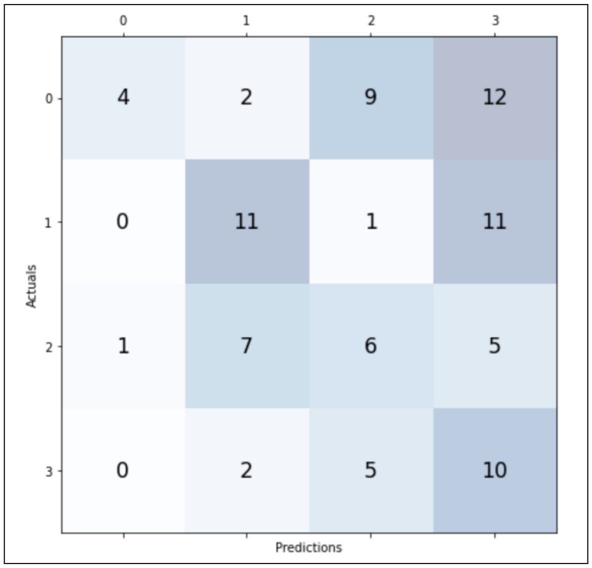

Fig. 13: Confusion Matrix for 2D Data

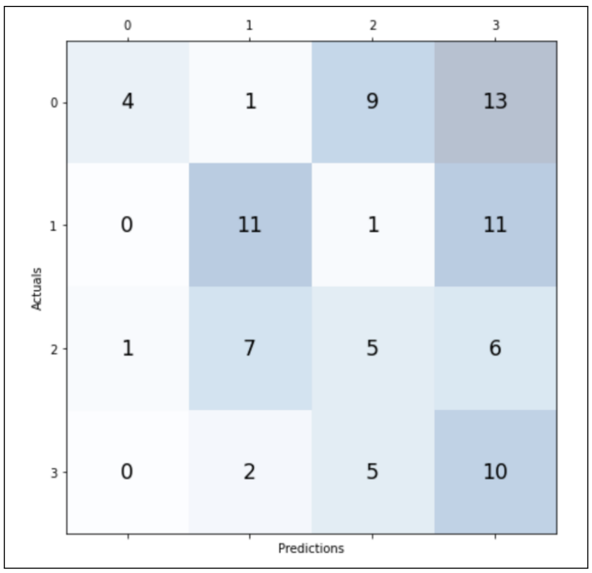

Fig. 14: Confusion Matrix for 3D Data

of which strings overlap. From these preliminary results, one observes that adjacent strings have increased overlap. This finding aligns with expectations since strings that exist close to each other spatially likely activate similar muscle fascicles during play. To investigate this approach, the team will now employ linear discriminant analysis (LDA) on the style data.

V. Linear Discriminant Analysis by Style

To decrease the computational load and reduce the risk of overfitting, the team begins by once again windowing the data and conducting PCA. The style data is then projected onto the first three principal components.

LDA works by finding the Fisher vector along which the projection of the data is best separated by class. To do so, LDA calculates the within-class and between-class variation. Computation with these values enables the generation of such a vector. Each data point can then be projected onto this vector,

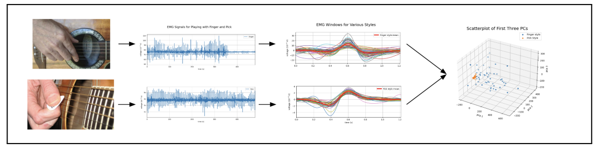

Fig. 15: Processing Style Data

as seen in Figure "EMG Projections on the LDA Normal Vector." The distribution of such points is also illustrated in the Figure "Histogram of Fisher Projections." There, one can clearly see distinct categories for the pick and finger data.

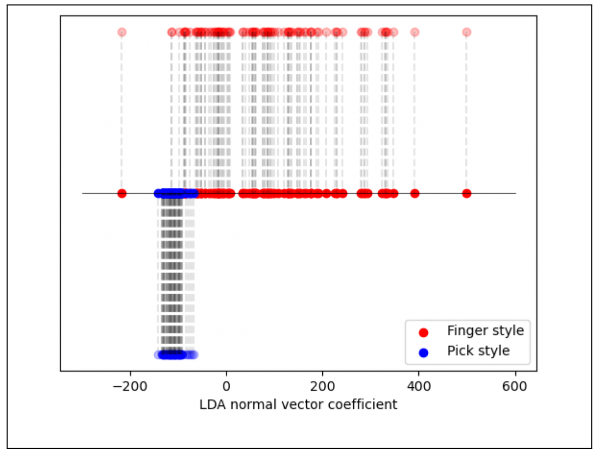

Fig. 16: EMG Projections on the LDA Normal Vector

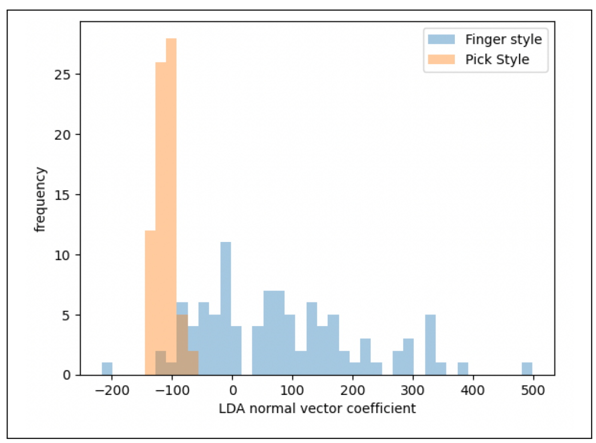

Fig. 17: Histogram of Fisher Projections

Along this vector, one can assign a decision boundary such that points below it are assigned to one class while points equal to or above it are assigned to another. While this boundary can be determined following projection, such a method risks overfitting the data given its employment of a posterioi information. To minimize this risk, the team employs a decision boundary that is equidistant to the class means of the projections. Notably, the pick data form a dense cluster while the finger data are spread out. This result makes sense given the reduced amount of motor activation required to move a plectrum compared to picking a string. The increased movement necessary for the latter tasks explains the more significant variability in the finger data.

To develop a holistic sense of LDA's performance, the team randomly separates the data into training and testing sets. Under each evaluation, 80% of the data is employed for training while the remaining 20% is involved in testing. Such a methodology aims to see whether overfitting occurs. The result of these simulations can be seen in the ROC curve. This curve treats detecting finger style as positive. Notably, the team identifies that the false positive rate (FPR) in all cases is 0%. This result arises from the high density of the plectrum data. In contrast, the true positive rate (TPR) experiences a degree of variability yet consistently is above 55%, indicating a high level of predictive power in the model. The TPR can likely be increased by moving the decision boundary toward the projected pick mean. The concern of this approach is that given the high density, the FPR will experience a rapid increase for marginal increases in TPR following a certain point. Placing the decision boundary right before this cliff would have significant implications for overfitting.

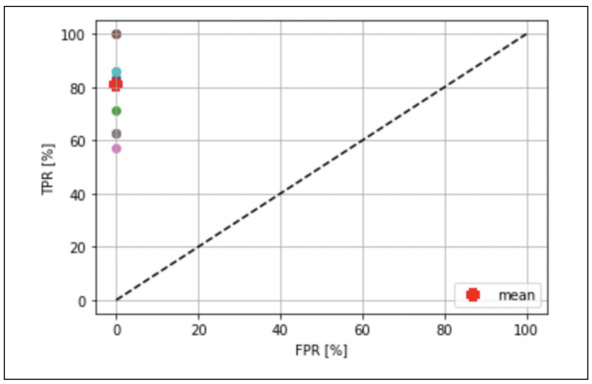

Fig. 18: ROC of LDA via Training and Testing

VI. Alternative Comparator

While LDA can be used to accurately determine if a note has been played fingerstyle or with a plectrum, this limits real-time performance. More complex calculations take more time, so a 'brute-force method of checking against thresholds will be attempted.

As noted in the windows, there is a difference in the amplitude between notes played by fingers or with a pick: fingerstyle notes have an average amplitude of $12\dot{1}0^{-4}$ mV, while pick-style notes have an average amplitude of $2.5\dot{1}0^{-4}$ mV. These values are obtained after removing the mean from each sample after noise above 15Hz has been removed. The output from this process should determine 2 things: whether a note has been played and the playing style of that note. Using constant thresholds is a static way to approach this problem.

The analog implementation would have 3 main stages. First, filtering noise above 4Hz. Filtering at this point preserves the shape of the EMG and maximizes SNR. Second, removing bias from the EMG signal. This value can be determined by averaging all of the recorded data thus far. The third is a comparator for the threshold at which a note is considered picked and a higher threshold for a note to be considered being played fingerstyle. As the data was filtered, the next two stages can be simulated as below: The threshold for fingerstyle is set at $4\dot{1}0^{-4}$ mV, while the pick is set at $1.5\dot{1}0^{-4}$ mV. This conveniently matches the previous windowing thresholds, but the first threshold is set by no picked samples having an amplitude greater $4\dot{1}0^{-4}$ mV.

As 40 seconds of recording from both playing styles have been captured, we expect to see 20 samples (1 sample every 4 seconds). This idea is confirmed by visual inspection of the plots.

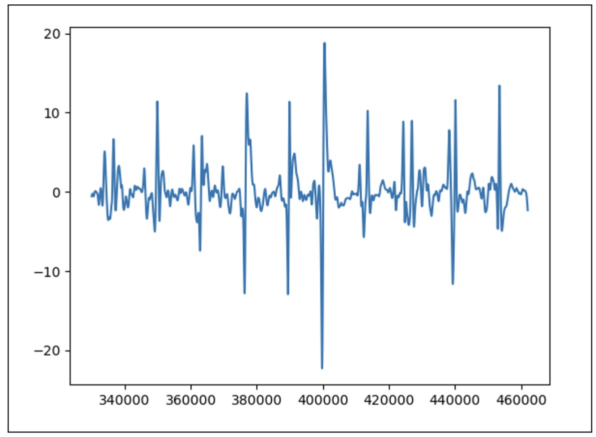

Fig. 19: Alternative Comparator: Finger Data

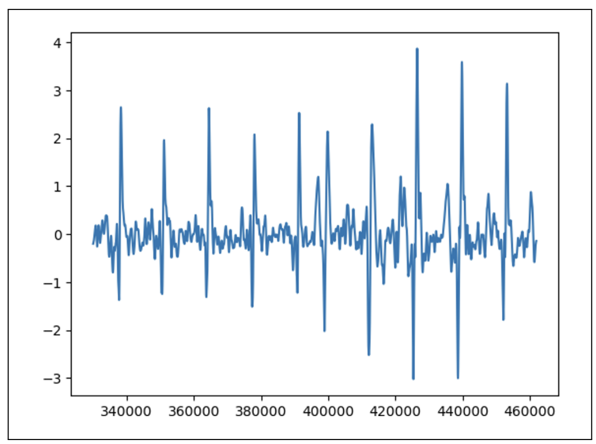

Fig. 20: Alternative Comparator: Pick Data

Visually, this represents 2 issues with using static thresholds for determining whether a note has been played. For fingerstyle notes, noise in between samples is a greater amplitude than the picked samples. Additionally, filtering was attempted to solve this issue, but it showed no positive change.

However, both sets of samples have a similar signal-to-noise ratio, meaning comparable thresholds can be used for both if scaled to -1,1. This solves the issue of determining a note from noise but creates a new issue as the playing styles can no longer be distinguished from each other by amplitude. Consequently, the collected data only facilitates distinguishing a note from noise or distinguishing a picked note from a note played fingerstyle.

VII. Conclusion and Future Directions

This paper has successfully investigated the efficacy of unsupervised and supervised techniques when applied to muscle data. Specifically, the authors were able to employ standard filtering, peak finding, and dimensionality reduction techniques to enable the prediction of guitar strings and style. The authors have observed that the supervised technique had significantly improved performance, likely because of cleaner data and the application of data classification during the labeling process. These efforts have shown that PCA, K-means clustering, and LDA are all effective tools when analyzing EMG signal data. The steps outlined in this paper provide a foundation for future scaling and development.

The efforts undertaken in this paper are limited due to the low-quality sensor employed. A more precise sensor or an array of sensors would enable more fine data that would improve the ability of the mathematical methods employed to distinguish the classes. Further investigation of parameters like window size, filter settings, and frequency data would enable a more thorough understanding of existing data and its capacity to provide insights. Adjusting the window size could enable a more thorough understanding of the temporal dynamics of guitar play, while the frequency domain may offer nuances not obvious in the time domain.

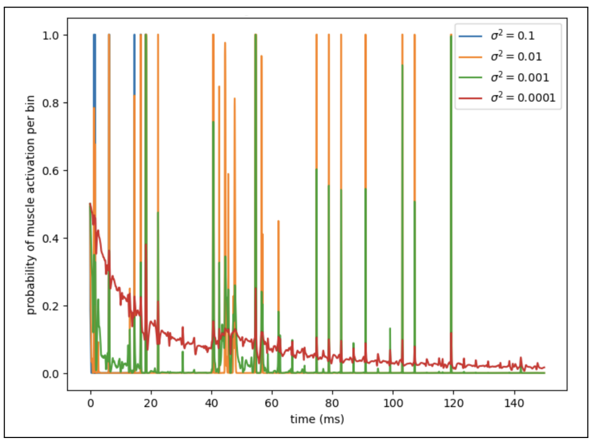

Fig. 21: Binary Filter of EMG Data

Another future endeavor is the application of independent component analysis (ICA). The use of multiple sensors would provide the data throughput required to conduct ICA, which can isolate the activity of distinct muscle fibers, thereby increasing model efficacy. Additionally, using a binary filter could provide insight into the activation of individual muscle fascicles.

Future steps remain to find ways to obtain more granular data and use this to develop a real-time process that responds instantly to the user's movements. Such an undertaking will enable more efficient guitar performance and increased accessibility.

VIII. Acknowledgement

The authors would like to thank the ES157 course staff for their help in the development of this manuscript.

IX. References

[1] C. Amma, T. Krings, J. Böer, and T. Schultz, "Advancing Muscle-Computer Interfaces with High-Density Electromyography," in Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, New York, NY, USA, Apr. 2015, pp. 929–938. doi: 10.1145/2702123.2702501.

[2] F. Mazhar, "Automatic Guitar Chord Detection," 2012, Accessed: Dec. 05, 2022. [Online]. Available: https://trepo.tuni.fi/handle/123456789/20923.

[3] M. Stein, J. Abeßer, C. Dittmar, and G. Schuller, "Automatic Detection of Audio Effects in Guitar and Bass Recordings," presented at the Audio Engineering Society Convention 128, May 2010. Accessed: Dec. 05, 2022. [Online]. Available: https://www.aes.org/e-lib/browse.cfm?elib=15310.

[4] J. E. Tamani, J. C. B. Cruz, J. R. Cruzada, J. Valenzuela, K. G. Chan, and J. A. Deja, "Building Guitar Strum Models for an Interactive Air Guitar Prototype," in Proceedings of the 4th International Conference on Human-Computer Interaction and User Experience in Indonesia, CHIuXiD '18, New York, NY, USA, Mar. 2018, pp. 18–22. doi: 10.1145/3205946.3205972.

[5] H. Touyama and M. Mizuguchi, "Decoding of Hand Shapes Based on ElectroMyoGraphic Signals during Playing Guitar Chords," in HCI International 2011 – Posters' Extended Abstracts, Berlin, Heidelberg, 2011, pp. 197–200. doi: 10.1007/978-3-642-22095-1_41.