**Auxiliary Information:**

Members:

Jared Frazie:

Department of Computer Science

Middle Tennessee State University

Murfreesboro, Tennessee USA

jf5s@mtmail.mtsu.edu

Romario Nan:

Department of Computer Science

Middle Tennessee State University

Murfreesboro, Tennessee USA

rrn2h@mtmail.mtsu.edu

Ethan Lawing:

Department of Computer Science

Middle Tennessee State University

Murfreesboro, Tennessee USA

eml3y@mtmail.mtsu.edu

Deven Kennedy

Department of Computer Science

Middle Tennessee State University

Murfreesboro, Tennessee USA

dak3s@mtmail.mtsu.edu

1. **Abstract**

A Neural Network with a LSTM function that will take a piece of music and create a new one. The new one will have some characteristics for the original piece but at the same time, it will be different. We will be using mainly videogame music. The model will be using a MIDI file. The model will make the music into chords by decomposing it using music21. It will be using a pattern of 16 notes. The notes will be represented in a 2D array shape with only 0 and 1 inputs So NNs works with it. The model at the end will combine all the nodes in a MIDI file.

1. **Introduction and Background**

Can the machine be a force of creativity? Can it take the very heat of music and learn from it, to create something new? These were the questions asked in the conception of this project. Our network is a generative model. It has been trained using examples, which it then attempts to make something akin to, but not identical to. The more complex and interesting original ideas can be generated by neural networks, the wider of an application they will have. If neural nets can write music, then what can’t they do? Such is our goal. We aim to train a Neural Network that writes original music.

1. **Methods**
2. Our examples used to train the model are music from Nintendo video games. They were then scraped and reformatted into MIDI files.
3. -------------------- Performance Metrics Section --------------------
4. We will be using an evaluation toolbox developed by Li-Chia Yang and Alexander Lerch [2]. This toolbox has an immense array of metrics to use that can be classified as either an objective or subjective metric. The subjective metrics are specialized for comparing two sets of data (i.e. comparing generated data and test data). The most notable concepts of these metrics are the computations of the intra-set and inter-set distances, the Kullback-Leibler Divergence (KLD), and the overlapped area (OA).
5. In order to find the two different types of distances a “pairwise exhaustive cross-validation is performed for each feature.” [2] I will also quote Yang and Lerch to explain what intra- and inter-set distances are: “If the cross-validation is computed within one set of data, we will refer to it as intra-set distances. If each sample of one set is compared with all samples of the other set, we call it the inter-set distances.” [2] The KLD and OA are similarity measurements. In order to explain what they show, imagine we have sets of generated data from two models, M1 and M2, and the training data. We calculate the intra- and inter-set distances for each set of data. Then we compute the KLD and OA for the inter-set distances of the data for M1 and the training, and we do the same for M2’s data and the training data. The M1/training KLD was smaller than the M2/training KLD. The M1/training OA was larger than the M2/training OA. This implies that the generated data from M1 is more similar to the training data than the generated data from M2. If all the data were music, that means the music M1 generated is more similar to the training music than the music from M2. That could mean the music from M1 is more human-like than the music from M2.
6. The objective aspect of this toolbox takes advantage of the varying parts of music that make a full song. Take for example the pitch count and note count for a song. The pitch count is the total number of different pitches used in a song. If a song used the pitches C, F, and G throughout the entire song, it would have a pitch count of three (**testing will help check if it actually comes out like this)**. The note count is the total number of notes used. If that same song used C four times, F one time, and G three times, it would have a note count of eight. Now that we have these varying musical structures, we can perform the computations mentioned above on specific aspects of songs and visualize more accurately where different data sets are outperforming others, or possibly where they both struggles.
7. **Results**

We are getting a 16 notes musical MIDI Sequence (.mid) file which containing the generated music.

1. **Figures**

**Diagram

Description automatically generated**

**Graphical user interface

Description automatically generatedGraphical user interface

Description automatically generated with low confidence**

1. **Discussion and Conclusion**
2. **Future Work**

Future work will include training the model for a longer time to see if the results will improve. We could also expand the model size to see if a larger size has better results. We also could run the model a few extra times to see if we will get better results. More runs will give us a better average for the model which will help notice if we need to fix any flaws. We could also use WaveNet for the model instead of LSTM. WaveNet is a system developed by Google to generate data from original data. The new data will be different from the original one. WaveNet is a Generative model. It is great for audio samples because we feed it a piece of music and it produces a similar piece [1]. Other Models online use drop-out layers which seems to help train the model and prevents over-fitting. Other scientists tried to make similar models and they used few LSTM functions to make their models. They also add few activation functions like ReLU and SoftMax. Early stopping will also help to prevent overfitting which had happen in few runs. Finally, we try to feed it other music and see the results.

1. **References** [1] Oord, Aaron van den, et al. "Wavenet: A generative model for raw audio." arXiv preprint arXiv:1609.03499 (2016).

[2] L.-C. Yang and A. Lerch, "On the evaluation of generative models in music," Neural Computing and Application, vol. 32, no. 9, p. 12, 2018.