iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** XII **Month of publication:** December 2022

DOI: https://doi.org/10.22214/ijraset.2022.48200

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue XII Dec 2022- Available at www.ijraset.com

Music Generation Using Recurrent Neural Networks

Atharva Joshi[1], Atharva Gokhale[2], Arnav Khandekar[3], Aalok Kesarkar[4], Shaunak Khade[5], Prof. Noshir Tarapore[6]
[1, 2, 3, 4, 5, 6]*Dept of Computer Engineering Vishwakarma University Pune, India*

Abstract: ***Over the past years music has been continually evolving through the its tempos and beats as well as it's melody. Traditionally, music is produced by a group of musicians with different instruments combining together to create a final synchronised product. In recent years, harmonies and beats were always considered to be generated manually. However, with the advent of digital technologies and software, it has become possible for machines to generate music automatically at an alarming pace. The purpose of this research is to propose a method for creating musical notes using Recurrent Neural Networks (RNN), specifically Long Short-Term Memory (LSTM) networks. To implement this algorithm, a model is created and the data is represented in the form of a percussive instrument digital interface (MIDI) file for easy access and interpretation. The process of preparing the data for input into the model is also discussed, as well as techniques for receiving, processing, and storing MIDI files for use as input.To enhance its learning capabilities, the model should be able to remember previous details of a musical sequence and its structure. This paper discusses the use of a layered architecture in the LSTM model and how its connections interweave to create a neural network.***

Index Terms: MIDI, RNN, LSTM, Music Generation.

I. INTRODUCTION

Recent research and development in music technology have made it possible to replicate artists using deep learning and similar technologies. These AI models are referred to as Generative Adversarial Networks or GANs. They learn howto perform complex tasks as they process large data sets. Music does not have a set dimension because it is a series ofnotes and chords. Traditional deep neural network approaches cannot be used to make music because they presume fixed di- mensionality of inputs and targets/outputs and independence of outputs. As a result, it was evident that a domain-independent Identify applicable funding agency here. If none, delete this. approach that learns to map sequences to sequences would be beneficial. Music is an art that employs the science of harmony of the pitch to achieve the beauty of sound. It is frequently used asa form of artistic expression and may be heard without theuse of any non-musical instruments. The term 'note' refers toa single unit of musical measurement, denoted by the sign ''. With or without the presence or addition of other notes, notes might behave differently. Music is organised in the form of notes, chords, or even a scale. Music has its sense to it sinceit brings pleasure to the ear. It can be frustrating since thereare different genres of music out there.

The genetic algorithm is a method for creating music by building upon existing compositions. Evolutionary algorithms may emphasise the powerful rhythm in each fragment and merge them into separate pieces of music. However, it isinefficient since each iteration step includes a latency. Addi- tionally, it is difficult to obtain coherent and detailed rhythmicinformation due to the lack of context. To solve the preceding scenario, we need a system that can remember the previous note sequence, forecast the following one, etc. Recurrent Neural Networks, namely Long, and Short-Term Memory RNNs, are deployed.

We presented an algorithmic approach to music generation. Learning musical genres from any current instrumental beat to beat rhythm may be used to generate a new music pattern. Thissensitive objective demands the system's ability to apprehend previous knowledge to extract musical elements and project future musical style patterns. Furthermore, the system must learn and alter the original musical patterns.

This study presents an algorithm based on LSTM networks (a type of neural network) that can be used to autonomously generate music and melodies without human intervention. Themain goal is to develop a model that can learn from a series of musical notes, analyze them, and then generate a set of high-quality musical sounds.

We expand the use of RNNs to create a music gener- ator by combining character-based neural language models such as Character RNN and LSTM cells , which include bi-directionality and attention.

The ABC notation is used to express music in text format. Other more expressive audio for-mats, such as MIDI and Mel Frequency Cepstral Coefficients (MFCCs), exist, however they are primarily sound-based and may not be transcribable to sheet music.

II. LITERATURE REVIEW

We introduced an algorithmic approach to music synthesis. Generating musical genres from any current instrumentation beat to beat rhythm would be used to build a new musical pattern. This crucial aim involves the system's ability to retrieve foreknowledge to extract musical elements and project prospective musical style patterns. Secondly, the system must learn and alter the original musical patterns.

A. Music Generation with Variational Recurrent Auto-Encoder

A new architecture combining an artificial neural network with a variational auto-encoder supported by historical data and filtering heuristics has been developed to generate pseudo- live, aesthetically pleasing and melodically diverse music. This is the first application of VRASH to music generation, and it achieves a good balance between global and local structurein the track. The proposed structure is relatively simple to implement and train, and it enables control over the style of theoutput, generating tracks that match the specified parameters.

B. Deep Recurrent Music Writer: Memory-enhanced Varia-tional Auto encoder based Musical Score

The authors developed a new metric to evaluate the quality of generated music and used it to assess the outputs of a Variational Auto-encoder-based generative model for au- tomated music composition. They applied this measure to systematically and automatically adjust the parameters andarchitectures of the generative model to optimize the musical outputs in terms of their similarity to a particular musical style.

C. Towards Music Generation with Deep Learning Algo-rithms

The researchers developed a multi-layer Long-Short Term Memory (LSTM) Recurrent Neural Network (RNN) and afeed-forward network based on a collected data set, as well as an LSTM-based Encoder-Decoder architecture as a baseline model. However, the models did not achieve the goal of generating a 60-second-long sequence of polyphonic music. The limitations of the models were discussed, and it was determined that further refinement is needed before being ableto generate actual musical sequences.

III. PROBLEM STATEMENT

Deep Learning is already becoming extremely prevalent, with purposes relating to computer vision to natural language, medical, voice recognition, generating art, adding sound to silent films, translation software, marketing, and self-driving vehicles. We will apply deep learning networks to producing music. Music does not have a set dimension since it is aseries of sounds and chords. Conventional neural network- based algorithms cannot be used to make music because they presume fixed dimensionality of input and benchmarks and dependence of outputs. As a result, a methodology that learns to map sequences to sequences would have been beneficial. We intended to express how Recurrent Neural Networks can be used to compose music (RNN).

A Long Short-Term Memory (LSTM) neural network. Whether handmade or synthesised, music production is a notoriously challenging feat due to the multiple components. Considering this, we briefly summarise the intuition, theory, and implementation of LSTMs in music generation, createand introduce the network we discovered to achieve this goal, identify and address issues and challenges encountered, and include potential future improvements for our network.

IV. OBJECTIVES

Our system aims to create a model that can compose melodies and rhythms automatically without human input. The model is able to remember previous data descriptions and generate harmonious music using a single-layered LSTM model. It is capable of learning harmonic and melodic note sequences from MIDI files of piano notes.

We want to investigate the effect of increasing the number of LSTM units and testing different hyper-parameter values on the model's performance. We believe that far more studies could further optimise this model using a substantial number of computations.

The primary objective of autonomous music creation isto aid musicians in composing and refining music. Another important consideration is the amount of oversight used for theprototype. There is total autonomy and mechanisation at one end of the scale with no human input.

It might also be more interactive, with early stops included in the model to monitor the music generation process. This paper's neural network technique is designed to be non-interactive. The MIDI file format is also designed for this dimension since it providesa full final result that is machine-accessible without human contact.

The degree of autonomy is an intriguing perspective advancement for genuine musicians who can rupture the sys- tem in the middle of content creation. Flexibility in user input is another important fundamental objective. Our architecture is adaptable: the user could change the number of layers,hidden layers size, sequence length, time steps, batch size, optimisation algorithm, and learning rate. Users may further adjust the window of notes fed into the note-axis LSTM model as well as the number of time steps input into the model.

We can produce high-quality music, but there is still consid-erable room for improvement. For now, the music is created by only one instrument. It would be fascinating to hear what music the model could produce if taught to play multiple instruments. A technique that can handle unfamiliar notes in music may be added to the model. By filtering unknown notes and replacing them with known notices, the algorithm can generate more robust quality music. We want to use this approach in the future to produce mood-based music basedon user input. Music has been seen in studies to aid people with dementia and Parkinson's. In conclusion, this system mayproduce music based on the patient's demands.

V. EXISTING SYSTEM

It has frequently been suggested that a musical moment maybe defined by an emotional influx and outflow of energy. Con-sidered a universal quality of music, tension may be influencedby a variety of musical elements, including pitch range, noise level tempo, note density, harmonious relationships, and latentdemands.

The music system employs a multi-level biomimetic design that informally separates actions from processing (adaptive mappings by the RL algorithm) and sensing (the reward function) (changes of musical parameters). This indicates a de-velopment above the typical separation of sensing, processing,and response paradigm that was at the foundation of earlier artificial intelligence models.

The musical agent must select a series of musical gestures (musical parameters) that will heighten the tension heard by the spectator.

The supervised learning algorithm must select what musical action to take based on the emotionaldata provided by the listener in real-time in order to solve this active reinforcement learning (RL) challenge.

VI. PROPOSED SYSTEM

The LSTM layer of a recurrent neural network can eitheroutput a sequence or a matrix when given a sequence as input.Dropout layers are a way to prevent over-fitting in a neural network by randomly setting a fraction of the input units tozero during each training update. The fraction of units thatare set to zero is controlled by a parameter. Dropout helps toregularize the model by reducing the reliance on any one input unit. Dense layer, also known as a fully connected layer, is a type of layer in a neural network where each input node is connected to every output node. This means that the outputof each node in the dense layer depends on all the inputnodes. Dense layers are commonly used in many neural network architectures and are responsible for learning complexrelationships between the input and output data.

The activation layer determines which activation function the neural network will use to calculate the output of each node.

The first parameter for LSTM, Dense, and Activation layers determines the optimal number of nodes in each layer. The first parameter for a Dropout layer specifies the percentage of input units that should be set to zero during training to prevent over-fitting. This parameter helps to regularize the model and improve its generalization ability.

An unique parameter called as input shape needs to be pro- vided for the first layer. The parameter's purpose is to provide the network with information about the type of training data set. Our system has variety of outputs which should always be represented by the same number of nodes in the final layer.

By using an output layer with a specific number of nodes that corresponds to the number of classes in the data-set, we can ensure that the output of the network will match the classesin the data-set.

We use two LSTM layers, three Dropout layers, two Dense layers, and one activation layer in this system.

We will utilise categorical cross entropy to determine the loss for each iteration of the training because each of our outputs only belongs to one class and we have more than two classes to work with. The network will be trained for 200 epochs (iterations) in our lesson, with 64 samples being sent through the network in each batch.

We use model checkpoints to ensure that we can stop training at any time without losing all of our work. Model checkpoints allow the weights of the network nodes to be stored to a file after each epoch. We can stop running the neural network without worrying about losing the weightsonce the tests are satisfied with the loss number. Otherwise, we'd have to wait until the network had gone through all 200 epochs before we could save the weights to a file.

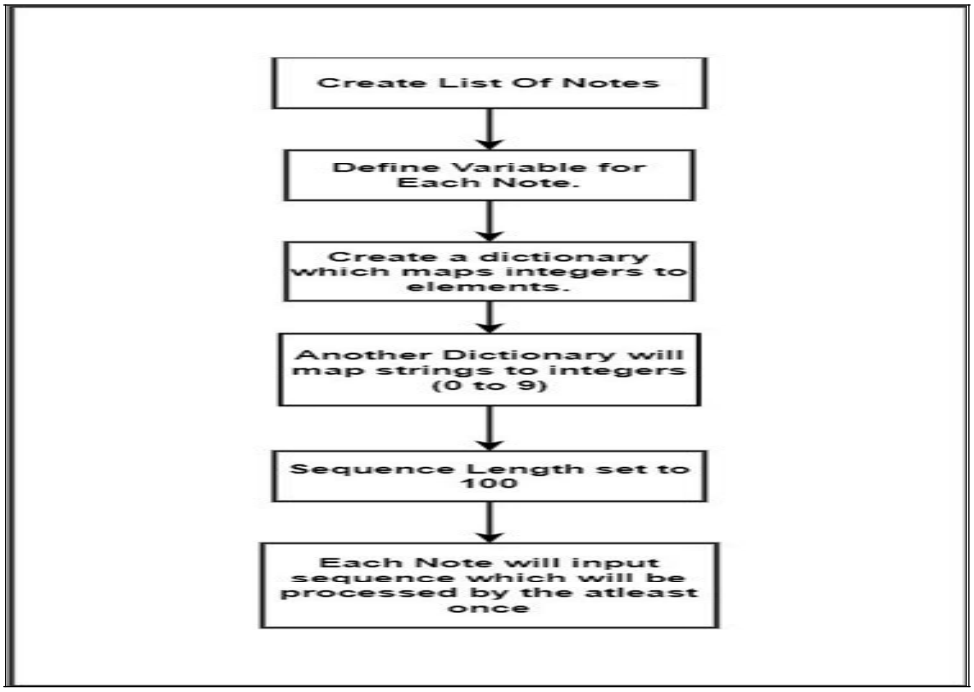


Fig.1. Basic Flow Diagram

Fig(1) : The basic flow of the proposed system isdepicted in Fig (1). The first thing the solution does is generate a list of notes derived from the training MIDI files. Following that, it defines a variable for each note. A dictionary is created, which assigns unique integers to all notes across all octaves. Following that, a new dictionaryis created that maps strings to previously declared integers ranging from 0 to 9. After this, a 100-note sequence is created.

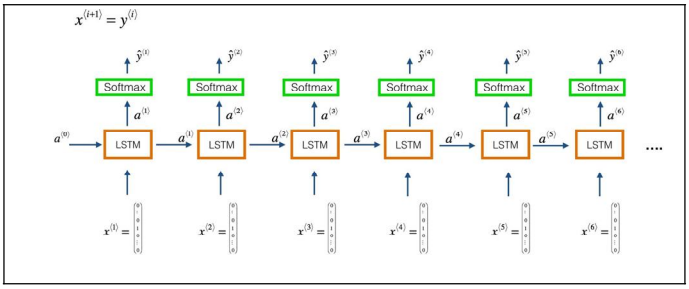


Fig. 2

Fig(2) : To train a model, we adopt a character level-based architecture. So, at the present time step 'tt', each input notein the sound file is used to forecast a next note in the file,i.e., each LSTM cell accepts the previous time step activation (at1) and the previous layer actual output (yt1) as input. The following diagram depicts this (Fig 2).

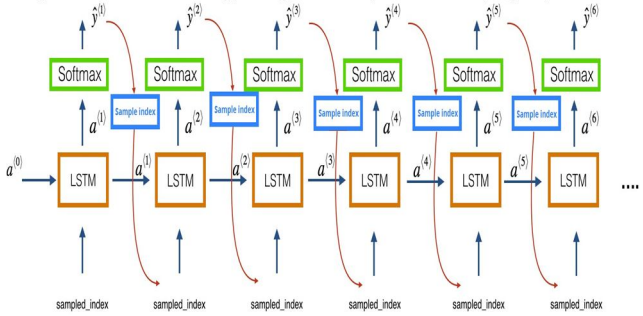


Fig. 3

Fig(3) : At each sampling step, the activation and cell state from the preceding LSTM cell will propagate into the following cell, where they will be used to generate a fresh output.

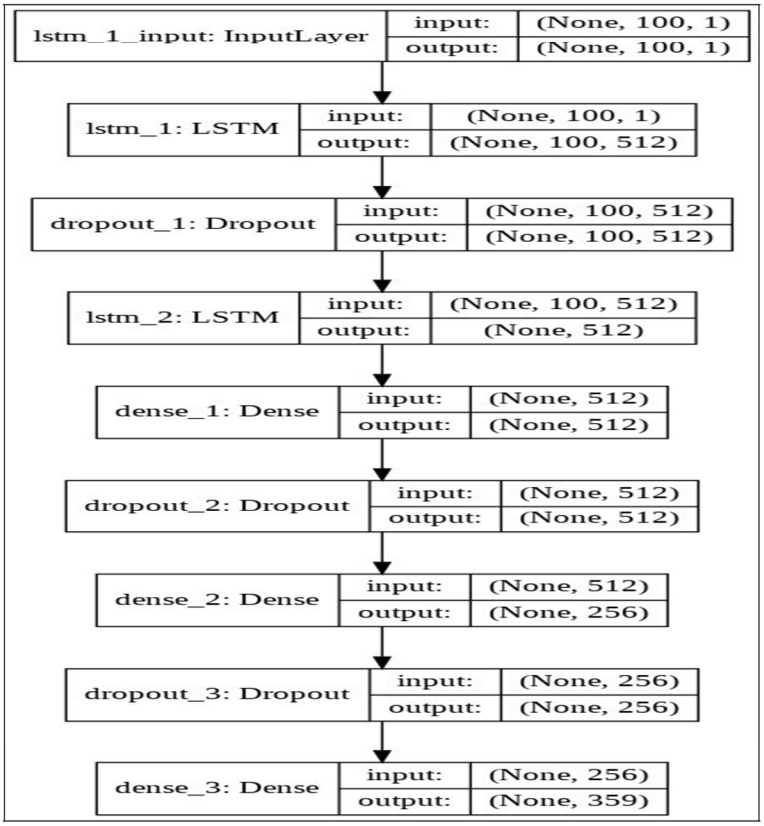


Fig.4

Fig(4) :A diagram of the model and its layers. We developedan LSTM model that learns to predict the next note in a fixed- length sequence of notes. A plot of the model layers is shown above.

VII. PROPOSED SYSTEM REQUIREMENTS (HW/SW)

The RNNs are one of the Neural Networks' versionsor alternatives. An RNN model (Recurrent Neural Network Model) is a form of neural network which is most typically used for Natural Language Processing, Speech Recognition, and Composition. A conventional neural network processesan input via several layers. It produces output under the assumption that two successive inputs are independent of one another, but RNN adds another parameter: memory.

Our data-set includes piano recordings in MIDI format. A MIDI file is more compact, easier to alter, and can be con- verted to any instrument because it provides player informationsuch as 'note on' and 'note off.' These files will need training.Third party python libraries like Keras, TensorFlow, numpy will be used to adjust and train data. We will be employinguse of Keras to build our model architecture and train this massive amount of data. To pull information from these MIDI samples, we will deploy the music21 toolkit.

To effectively train on large datasets, high-performance graphics processing units (GPUs) such as the GTX 1080 (with 8 GB of video RAM) may be required. These GPUs can process an average of 14,000 instances per second and can perform convolutional neural network (CNN) or recurrent neural network (RNN) operations on a batch of 128 or 256 images in a short amount of time. However, these GPUs havea high power consumption of approximately 250 W and may require a full computer system, which can add an additional 150 W of electricity for a total of 400 W.Therefore deploying and training the datasets on Cloud-based series like Google Cloud is optimal and efficient.

VIII. RESULTS

Models have learned some of the structural rules of the transcriptions in the data-set, as well as stylistic conventions. Results observed were as follows.

1) Each of the measurements in these cases is appropriatelytallied (but this is not always the case)

2) A transcription may have two 8-measure portions that are repeated twice.

3) A segment may have a second ending that differs from the first.
4) A segment concludes with the tonic. measure beforethe conclusion of a section, the dominant appears. Re- gardless, these transcriptions aren't "session-ready" rightaway. Our present research is at enhancing such models by including domain expert adjustments to the output.

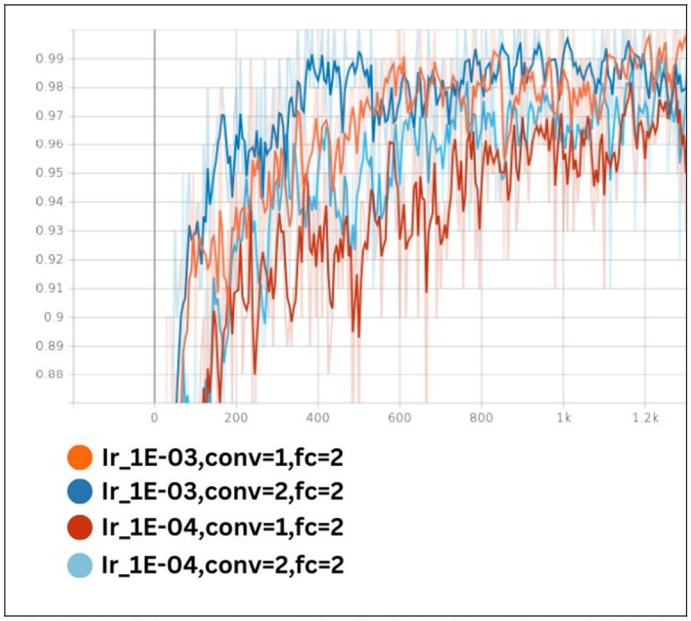


Fig. 5. The training accuracy after 800 epochs on Piano MIDI files

Fig(5) :A representation of the weight distributions at various stages of the LSTM layer used in this model. LSTM/recurrent kernel describes weights for previous state and time saved in order to provide the ability to recall earlier details and the structure of the model.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
lstm_1 (LSTM)	(None, 100, 512)	1052672
dropout_1 (Dropout)	(None, 100, 512)	0
lstm_2 (LSTM)	(None, 512)	2099200
dense_1 (Dense)	(None, 512)	262656
dropout_2 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 256)	131328
dropout_3 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 359)	92263

Total params: 3,638,119
Trainable params: 3,638,119
Non-trainable params: 0

Fig. 6 Summary of the LSTM Model used.

Fig(6) : This chart shows the summary() method's results. Each row denotes a layer, and the names of the rows are distinct enough self explanatory.

a) The "Output Shape" column displays the shape of each output, which is the output of each layer. The output of each layer is used as the input of the first layer.
b) In the params section shows how many parameters havebeen taught for each layer.
c) The total number of parameters, which includes both trainable and non-trainable parameters, is displayed at the end of the model summary. This model's layers can all be taught.

IX. CONCLUSION AND FUTURE SCOPE

We built a neural network utilising the RNN (LSTM) modelto automate music synthesis. The LSTM model has provento be a good model for capturing long-term dependencies. Our network learned a musical style by handing it musicalnote objects, which were then used to generate music. Webuilt a prototype in which the algorithm makes music onits own; Extending on prior work, we organised the data to utilise one-hot encoding and provided various musical notes for the network. In the future, we want to develop our network to learn and utilise even more song properties, recognise song beginnings and ends, and support multi-track inputs and outputs.

This report also discusses improving model feasibility and previous and present prospects. The Million Song Data-setwill be used in future research to see how well this model scales. The model was designed with essential functional characteristics of flexibility and generalisability. The basic ideaand approach of algorithmic music training and creation were described.

Our neural network model and algorithm can be used to create a wide range of commercially viable products. Basedon their interests and environment, the system can be utilized to generate mood-based new music for people. Furthermore,it can quicken the process of music composition and people's artistic instincts. It can also be used to make Indian classical music based on the set rules of the 'Ragas'(A raga in Indian classical music is a set of melodic rules for improvisation, similar to a musical mode. It is an integral and significant element of Indian musical tradition and does not have anequivalent in Western classical music.)

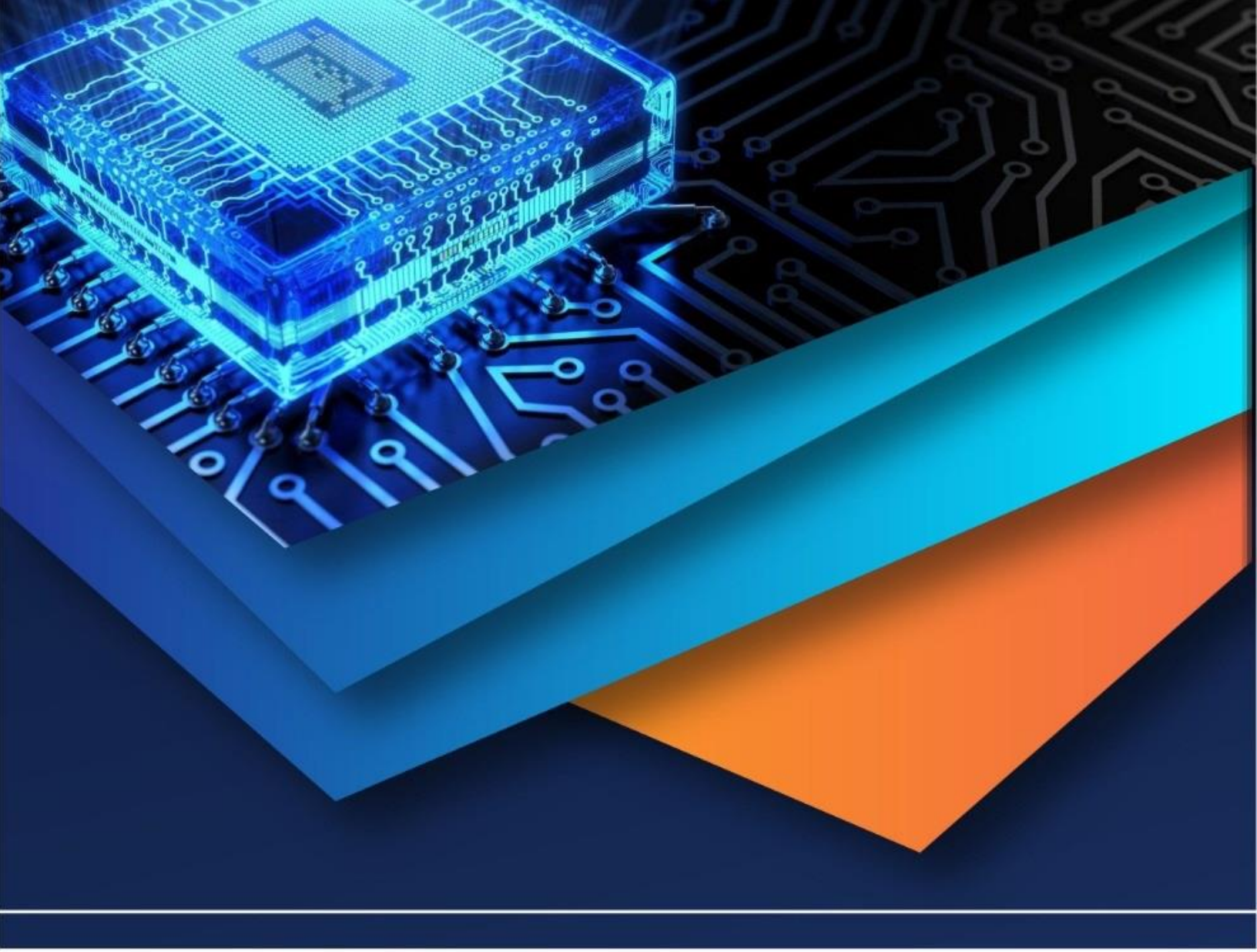
X. ACKNOWLEDGMENT

We thank Vishwakarma University's Faculty of Science andTechnology for providing us with this opportunity to present our study and research to the industry experts.

Last but not least, we would like to thank our connections from the industry who guided us, friends who supported us, and respondents for their active participation encouragement and willingness to spend time on our proposed system.

REFERENCES

[1] Agrawal, Shipra, and Navin Goyal. "Analysis of Thompson sampling forthe multi-armed bandit problem." In Conference on Learning Theory , pp. 39-1. 2012.
[2] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Modeling temporal dependencies in high-dimensional sequences: Appli- cation to polyphonic music generation and transcription. ICML, 2012..
[3] Douglas Eck and Jurgen Schmidhuber. A first look at music compositionusing lstm recurrent neural networks. IDSIA.
[4] Sigurur Skúli "Music Generation Using a LSTM Neural Network in Keras" https://towardsdatascience.com/how-to-generate-music-using-a-lstm-neural-networkin- keras-68786834d4c5 (2017). (Accessed on 16/10/22-22:00)
[5] Hochreiter, S., Bengio, Y., Frasconi, P., and Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In Kremer, S. C. and Kolen, J. F., editors, A Field Guide to Dynamical Recurrent Neural Networks. IEEE Press.
[6] P erez-Ortiz, J. A., Gers, F. A., Eck, D., and Schmidhuber, J. (2002). Kalman filters improve lstm network performance in problems unsolv- able by traditional recurrent nets. Neural Networks. Accepted pending minor revisions.
[7] Christopher Olah. "Understanding LSTM Networks." https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (2015) (Accessed on 16/10/22-22:00)
[8] Akshay Sood "Long Short-Term Memory" https://pages.cs.wisc.edu/ shavlik/cs638/lectureNotes/Long(percent- sign)20Short Term(percent-sign)20Memory(percent-sign)20Networks.pdf (2016) (Accessed on 16/10/22-22:00)
[9] PDillon Ranwala "The Evolution of Music and AI Technology" https://wattai.github.io/blog/music-ai-evolution (2020) (Accessed on16/10/22-22:00)
[10] Other piano data source: http://www.piano-midi.de/midi-files.htm (Ac- cessed on 16/10/22-22:00)
[11] https://github.com/vishnubob/python-midi (Accessed on 16/10/22-22:00)
[12] N. Rafal Jozefowicz, Ilya Sutskever, and Wojciech Zaremba. "An Empirical Exploration of Recurrent Network Architectures." (2015)
[13] P. Y. Nikhil Kotecha, "Generating Music using an LSTM Network," arXiv.org,vol. arXiv:1804.07300, 2018.
[14] T. a. H. G. Tieleman, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," COURSERA: Neural networks for machine learning, vol. 4, no. 2, pp. 26-31, 2012.
[15] D. P. W. E. B. W. a. P. L. T. Bertin-Mahieux, "The million songdataset,"International Society for Music Information Retrieval (IS- MIR'11), vol. 2, no. 9, p. 10,2011.
[16] "Musical-Matrices," 10 July 2016.[Online] Available: https://github.com/dshieble/Musical-Matrices/tree/master/Pop-Music- Midi.(Access ed on 16/10/ 22 17:00).
[17] C. B. Browne, "System and method for automatic music generationusing a neural network architecture," Google Patents, 2001.
[18] Michael Phi "Illustrated Guide to LSTM's and GRU's: A Step by Step Explanation." https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-sa-step-by-step- explanation-44e9eb85bf21 (2018)
[19] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Modelingtemporal dependencies in high-dimensional sequences: Applicationto polyphonic music generation and transcription," arXiv preprint arXiv:1206.6392, 2012.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY