

<sup>1</sup>Fei Wang

# Design and Implementation of Automatic Music Composition System Based on Deep Learning



**Abstract:** - Recently, there has been a notable increase in interest surrounding the utilization of Deep Learning Approaches for AMCS in recent times. However, current methods often lack coherence and authenticity in the generated music. This study proposes a new approach that utilizes RNNs to bridge the gap between traditional compositional methods and modern deep learning techniques. The goal is to produce expressive and coherent musical pieces. Our methodology involves designing and implementing a customized RNN architecture that can effectively capture the complex temporal dependencies present in musical sequences. We experiment with different types of including LSTM networks, RNNs, and GRUs, to overcome challenges such as vanishing gradients and better model longer-term dependencies in music. To train the neural network efficiently and improve model convergence, various deep learning optimizers are utilized in our system. Specifically, we use SGD optimizer to improve the hyper parameters of LSTM and GRUs. The data for training is converted into MIDI format and analyzed, with music lines being identified through a similarity matrix technique. The MIDI data is then prepared for use in the LSTM and GRUs networks. The resulting music is assessed using both objective measures, such as mean squared error, and subjective approaches. This research adds to the advancements in automatic music composition by demonstrating the capability of RNNs in capturing and producing complex musical arrangements.

**Keywords:** Deep Learning Approach, Automatic Music Composition Systems (AMCS), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, Gated Recurrent Units (GRUs) and Stochastic Gradient Descent (SGD) optimizer

## I. INTRODUCTION:

Music serves as a significant means for individuals to express their emotions and sentiments. It has been established as a set of principles for composing music. These principles serve as guidelines for creating music and regulate the standards for its composition. Generally, A full musical composition consists of three key elements: pitch sequence, music style and rhythm. The style of music often reveals itself through the employed chord progressions, while rhythm is defined by the repeating beats found in various sections. Central to the composition is the pitch sequence, which carries significant weight in shaping the overall piece. A musical composition can be divided into two parts: melody and accompaniment. The melody is responsible for conveying the main impression to the audience, while the accompaniment enhances harmony and adds flavor to the music [1]. Composing satisfactory music is a complex and challenging task as all these components and their interactions need to be taken into account. In recent years, there have been attempts to use artificial intelligence technology to analyze and create music, particularly in automatic accompaniment systems. While this approach has shown promising results, composing melodies still poses a major challenge due to the vast number of possible combinations of notes and beats.

In the realm of skilled musicianship, absorption encompasses a variety of experiences that may occur during practice or performance. Various experiences can span from the mundane and uninteresting, akin to a routine day at work, to moments where the mind wanders, and even to intense occurrences that transiently shift fundamental aspects of consciousness, such as perceptions of time, space, and self [2]. Recent dialogues on absorption in musical pursuits (such as composing, performing, or listening) have emerged from the realms of cognitive psychology and music psychology, often taking an ethnographic viewpoint. These accounts suggest that absorbed listening involves both a wealth of sensory input and heightened focus on the task at hand, while also allowing for the opposite states such as mental wandering, disengagement from the activity, and dissociation.

Recognizing musical instruments is a recognized and established task in MIR. Its objective is to discern the specific instrument being played in a given recording. This task holds relevance for a range of other applications, including source separation, automatic music transcription, developing music recommendation systems, and assessing music similarity. Moreover, it can prove advantageous for endeavors such as identifying a song's mood

<sup>1</sup> School of Art and Design, Hubei University of Technology, Wuhan, Hubei, China, 430068

Corresponding author's email: 15927113085@163.com

Copyright © JES 2024 on-line : journal.esrgroups.org

or genre. [3]. The task of instrument recognition presents two primary challenges: distinguishing instruments in recordings featuring a single instrument and identifying the predominant instrument within an ensemble recording. Analyzing recordings with just one instrument is generally regarded as straightforward, with many researchers achieving high accuracy levels (around 90%, akin to humans' ability to classify instruments based on a single sound). The latter challenge is more intricate as it necessitates discerning the dominant sound source within an ensemble recording, a concept that lacks a precise definition.

Sheet music plays a crucial role in the lives of not only students, but all artists. It helps to broaden their skills and allows them to reach their full potential. Additionally, sheet music serves as a means of communication between different musicians. It provides a structured way for musicians to share their compositions with each other, exchange ideas, and come to agreements [4]. While it allows for the sharing of one's own compositions, it also enables access to others' works with their permission. Although there may be other options available, sheet music remains the preferred medium for musical composition in the industry.

The process of creating music has traditionally relied on a composer's expertise in combining musical knowledge, emotion, and creativity. With advancements in computer technology, various music-related tools have been developed, mainly focused on editing techniques like arrangement and mixing. These programs allow for the separation and recombination of different parts of a piece, using the composer's specialized understanding of musical composition. As technology continues to evolve, experiments in musical intelligence (EMI) have utilized algorithms to analyze existing music and generate new pieces with altered styles [5]. However, this method requires a significant amount of time. There have also been attempts to incorporate mathematical models into music composition, such as the Illiac Suite created by the ILLIAC I computer using the Markov algorithm. With the impressive progress made in artificial intelligence across various fields, a trend has emerged towards employing neural networks to generate music of higher quality. Systems based on recurrent neural networks, such as Magenta, DeepJazz, BachBot, FlowMachines, and WaveNet, are being developed to assist composers in swiftly producing music [6]. However, most of these methods rely on note-based composition which can be limiting as melodies often change within bars. As a result, some systems are now utilizing bars as a basic unit of composition.

Recently, deep learning has been used to produce music by taking into account specific musical characteristics. For instance, the CONCERT program utilizes RNN to create melodies, while Deep Bach utilizes a LSTM neural network to compose both melodies and harmonies, which has proven to be more accurate in identifying melodic structures compared to RNN. The Song from the PI model also employs a multi-layer LSTM to simultaneously compose melodies, harmonies, and percussion [7]. Additionally, GAN, known for its success in image processing, has been applied in various studies for GAN-based music composition. One example is the use of C-RNN-GAN, where two LSTMs are used to build a GAN model for composing melodies. However, due to its reliance on note-based composition, the quality of the generated music is often lacking. To address this issue, our study proposes an improved GAN model for composing melodies. By introducing a GAN model that is suitable for bar-based encoding, we were able to overcome the limitations faced by C-RNN-GAN when working with note-based compositions. Furthermore, by using the TFIDF algorithm and implementing a filter based on shallow structural descriptions from our previous research [8], we were able to successfully differentiate between high-frequency melodic components and non-melodic elements in music and extract relevant data for training purposes. The main contribution of this research is discussed below.

### 1.1 Research Contribution:

- RNNs make it possible to generate music that goes beyond simply copying existing pieces. They have the capability to produce unique and inventive musical works, introducing fresh patterns, arrangements, and melodies that may not have been previously explored in traditional composition.
- With the use of RNN technology, AMCS offer a more efficient and time-saving method for creating music. These systems can quickly generate a wide variety of musical pieces, providing composers and musicians with a plethora of material to incorporate into their own work.
- By training on a diverse range of musical styles, RNN-based systems allow artists to explore different genres and experiment with blending influences from multiple genres.
- An effective AMCS system is developed using LSTM model which can able to achieve maximum accuracy and lower loss factors when compared with the earlier baseline methods.

## II. RELATED WORKS:

In [9], an algorithm combines audio and video features using multimodal deep learning, outperforming single modal classifiers in music emotion classification. Through a diverse music video dataset, experiments demonstrate its superior classification efficiency over traditional methods. In [10], an automated pipeline was described for conducting large-scale cultural transmission experiments using singing. This revealed the evolution of 3,424 melodies across 1,797 participants in the US and India, demonstrating how individual biases and social dynamics influence the emergence of musical structures observed across different cultures. In [11], a meta-analysis of 62 longitudinal studies was conducted, indicating that music training yields slight yet consistent positive effects on both behavioral and brain measures related to auditory and linguistic processing. These effects were found to persist regardless of the type of training or control group used, underscoring the potential benefits of music education for neuro behavioral enhancements. In [12], the study reveals that nasal breathing enhances feelings of relaxation and happiness while listening to music, compared to oral breathing, which shows a tendency towards more negative emotions. Musical features like consonance positively correlate with positive emotions, while song complexity correlates with negative emotions, shedding light on the influence of breathing pathways on emotional processing during music listening. In [13], a system has been developed to precisely identify chord shapes from extremely brief music clips, achieving an impressive accuracy of 99.47%. This was accomplished through the utilization of LSF-deltaS deltaG features and a classification model based on LSTM-RNN. This advancement aids in musical composition analysis, transcription, and automated background music generation. In [14], a system leveraging Deep Learning, employing multi-layered GRU cells for source separation and LSTM cells for chord estimation, resulting in enhanced accuracy of sheet music generation from songs. This advancement expands the system's capacity to separate multiple sources and improves the precision of chord estimation for musicians and enthusiasts alike.

In [15], a sensor network-driven audio retrieval and vocal music teaching system has been developed, improving accuracy and efficiency through the optimization of sensor placement and algorithms. This system offers real-time monitoring, thorough sound evaluation, and guidance, showcasing its effectiveness in both audio retrieval and vocal music instruction. In [16], convolutional neural network approach for musical instrument recognition, addressing both single instrument classification and more challenging polyphonic recordings. Staged training, building on monotimbral analysis, enhances accuracy, offering promise for improved instrument recognition in music information retrieval tasks. In [17], unique resting-state functional connectivity patterns among three groups: Improvising musicians, Classical musicians, and MMT controls. Notably, Improvisational musicians exhibit heightened connectivity between the DMN and ECN. This suggests varying cortical network organizations among musicians, delineated by their training backgrounds and improvisation abilities. In [18], the article has been retracted due to a violation of peer review standards, as it was partly accepted based on a positive review from an illegitimate reviewer with a false identity. Apologies are extended to the affected reviewer and readers for this deception in the submission process. In [19], concept of "mind surfing" to reconcile the paradox of musical absorption involving both focused attention and mind wandering, proposing that skilled musicians can simultaneously experience intense focus and free exploration while "surfing" on a "musical wave".

In [20], the integration of blockchain technology with the IoMusT has been referred to as "Blockchain-based IoMusT", highlighting its potential for decentralized, transparent, and efficient management of copyrights, royalties, and musical data in this emerging domain. In [21], the PIEC system considers music theory and imitates characteristics of human-composed music, resulting in compositions that exhibit melodic progression akin to human-made music, demonstrating effectiveness in generating satisfactory and human-like musical compositions. In [22], a music piece is described using the "trajectory of fifths," which improves music data mining techniques by examining the variability of the music signature over time. The trajectory of fifths is beneficial for assessing music tonality and genre classification, as validated by experiments demonstrating its applicability through statistical analysis. In [23], an extension has been made to a melody generation method, now encompassing the creation of rhythmic content derived from a coherence structure extracted from a template piece. This extension specifically targets bertso melodies. The method uses pattern discovery and ranking to create rhythmic coherence structures, resulting in coherent and perceptually pleasing bertso melodies as evaluated through listener perception and comparison with bertso features. In [24], deep music recommendation algorithm has been developed, which relies on dance motion analysis to achieve an accuracy of 91.3% in recommending music genres. This was

accomplished using LSTM-AE models, surpassing traditional methods and showcasing the potential of motion-based music recommendation systems.

In [25], leveraging the strengths of deep neural networks for automatic feature extraction, the system incorporates historical music interaction data of users, demonstrating feasibility and effectiveness through experiments. This neural network-based music recommendation system is tailored specifically for college students, enriching music education by modeling individual preferences and delivering dynamic, personalized recommendations. In [26], a heuristic approach employing genetic algorithms and Markov chain models is introduced for the automatic composition of ICM. This method addresses the distinct constraints of raga-based music, employing a greedy strategy to strike a balance between exploration and exploitation. The technique incorporates domain-specific genetic operators and an evaluation function grounded in music theory, resulting in the efficient generation of melodic sequences while maintaining the desired distribution of musical notes. In [27], an approach converts training data to MIDI format, extracts melody lines, and generates music evaluated by objective metrics and professional surveys, showing advantages in capturing tone, rhythm, and artistic attributes. The earlier research ideas are summarized in table 1.

**Table 1 – Earlier Research Summary**

Ref.No	Algorithm	Methodology	Advantages	Disadvantages	Performance	Efficiency	Accuracy	Features Used	Measurements
[9]	Multimodal deep learning	Combines video and audio features using MDL for music emotion classification	Enhanced classification efficiency, comprehensive features from audio and video.	Lack of exploration on other modalities, training complexity.	Outperforms single modal classifiers.	High	High	Audio and video features	Classification efficiency
[10]	Music evolution experiments	Large-scale cultural transmission experiments using singing to examine melody evolution.	Reveals individual biases and social dynamics' influence on musical structures.	Limited scope, potential biases in data collection.	Shows influences of biases and social dynamics on music evolution.	High	N/A	Melodies from participants	Musical structure analysis
[11]	Music training effects	To explore the impact of music training on auditory and linguistic processing.	Shows slight yet consistent positive effects on behavioral and brain measures.	Mixed results from individual studies, correlational evidence.	Highlights potential benefits of music education on neurobehavioral enhancements.	N/A	N/A	Music training programs	Behavioral and brain measures
[12]	Nasal breathing	Study on the effects	Nasal breathing	Limited focus on	Demonstrates	N/A	N/A	Nasal vs. oral	Emotions during

	effects on music	of nasal breathing on emotions during music listening.	enhances relaxation and happiness compared to oral breathing.	specific emotions, potential biases in study design.	influence of breathing pathways on emotional processing during music listening.			breathing	music listening
[13]	Chord shape identification	System for accurately identifying chord shapes from short music clips using deep learning classification.	Achieves high accuracy (99.47%) using LSF-deltaS deltaG features and LSTM-RNN.	Limited to short music clips, potential challenges with complex chords.	Aids in music composition analysis, transcription, and automated background music generation.	High	99.47%	LSF-deltaS deltaG features, LSTM-RNN	Chord shape identification
[14]	Sheet Music Generation using Deep Learning	GRU cells, LSTM cells	Enhances accuracy in sheet music generation	Limited exploration in music recommendation from dance motions	Improved precision and accuracy in sheet music	NIL	NIL	GRU cells, LSTM cells	Sheet music generation
[15]	Sensor Network-based Audio Retrieval	Sensor placement optimization, Real-time monitoring	Enhances accuracy and efficiency in audio retrieval	Limited exploration in music recommendation from dance motions	Effective real-time monitoring and sound evaluation	NIL	NIL	Sensor data, Audio features	Audio retrieval, Sound evaluation
[16]	Convolutional Neural Network for Instrument Recognition	CNN, Single and polyphonic recordings	Addresses single and polyphonic instrument recognition	Limited exploration in music recommendation from dance motions	Enhanced accuracy in instrument recognition	NIL	NIL	CNN, Polyphonic recordings	Instrument recognition
[17]	Resting-State Functional	Functional connectivity analysis,	Reveals cortical network	Limited exploration in	Heightened connectivity	NIL	NIL	Functional connect	Cortical network analysis

	Connectivity in Musicians	DMN and ECN	differences among musicians	music recommendation from dance motions	ity in musicians based on training			tivity, DMN and ECN	
[18]	Retraction of Illegitimate Peer-Reviewed Article	Retraction of an article due to violation of peer review standards	Retraction of an article due to violation of peer review standards	Data transmission accuracy is moderate	Ethical transparency and correction	Inappropriate peer review	Retraction notice	Peer review ethics	Retraction notice
[19]	Concept of "Mind Surfing" in Music Absorption	Phenomenological research, Cognitive psychology	Proposes a new concept to reconcile music absorption paradox	Limited exploration in music recommendation from dance motions	New framework for understanding music absorption	NIL	NIL	Attention, Mind wandering	Music absorption
[20]	Integration of Blockchain with IoMusT	Blockchain technology, Music management	Enhances management of musical data and copyrights	Limited exploration in music recommendation from dance motions	Decentralized, transparent management of music data	NIL	NIL	Blockchain, IoMusT	Music data management
[21]	Phrase Imitation-based Evolutionary Composition	Genetic algorithm, Phrase imitation	Generates coherent and melodic compositions	Limited exploration in music recommendation from dance motions	Effective generation of music with melodic progression	NIL	NIL	Genetic algorithm, Phrase imitation	Composition generation
[22]	MDM for Trajectory of Fifths	Trajectory analysis, signature of Music	Enhances music data mining with trajectory representation	Limited exploration in music recommendation from dance motions	Valuable insights into music tonality and genre	NIL	NIL	Trajectory analysis, Music signature	Music data mining

[23]	Automatic Composition of Indian Classical Music	Genetic algorithm, Markov chain	Addresses constraints of raga-based music in composition	Limited exploration in music recommendation from dance motions	Efficient generation of Indian Classical Music	NIL	NIL	Genetic algorithm, Markov chain	Composition of Indian Classical Music
[24]	Evolutionary Computation-based Music Composition	LSTM networks, Grey wolf optimizer	Generates music with coherence and authenticity	Limited exploration in music recommendation from dance motions	Accurate generation of melodic sequences	NIL	NIL	LSTM	Evaluation of generated melodies
[25]	Neural Network-Based Music Recommendation System	Music recommendation system using neural networks	Provides dynamic, personalized recommendations	Limited to recommendation system	Demonstrated feasibility and effectiveness in experiments	NIL	NIL	LSTM-AE models	Accuracy of genre recommendation
[26]	Genetic Algorithm for Indian Classical Music Composition	GA and Markov chain method for raga-based music composition	Efficiently generates melodic sequences preserving raga constraints	Complexity in parameter tuning	Efficient and authentic raga-based compositions	Efficient composition method	High fidelity to raga constraints	Genetic operators, Markov chain model	Composition evaluation, raga authenticity
[27]	LSTM-Based Music Generation with Data Analysis	LSTM network for melodic sequence generation with data analysis	Captures tone, rhythm, and artistic attributes of high-quality music	Requires MIDI data format	Authentic and high-quality music generation	Efficient feature extraction	High music attribute capture	LSTM networks, MIDI data	Feature capture metrics, subjective evaluation

### III. FUNDAMENTALS:

#### 3.1 RNN Networks:

RNN, are a specialized type of artificial neural network designed to effectively model sequential data, including time series, text, audio, video, and various other forms. These networks are especially beneficial for tasks that involve sequences of data as inputs and outputs, and where there is a correlation among the elements within the sequence. RNNs, feed-forward neural network is specifically designed for modeling in the temporal realm. What sets them apart is their capacity to transmit information across different time intervals. Their architecture includes a parameter matrix for connecting time steps, allowing for training and utilization of sequential input data. RNNs are trained to produce output by considering both current input and data from previous time steps. They are particularly useful for analyzing temporal datasets. At its fundamental level, an RNN has a straightforward design, with a feedback loop incorporated into the network that enables the retention of

information. This loop enables the network to store a form of "memory" of past inputs. The general structure of an RNN resembles the following:

- Input:  $x_t$  is the input, time ( $t$ )
- Hidden State:  $h_t$  hidden state at time ( $t$ ). it inform about the sequence seen up to time  $t$ .
- Output:  $y_t$  is the output at time( $t$ )

The RNN equations is summarized:

$$\begin{aligned}
 h_t &= f\{W_{hh}h_{t-1} + W_{xh}x_t + b_h\} \\
 y_t &= g(W_{hy}h_t + b_y)
 \end{aligned}
 \tag{1}$$

Here, Weight matrix is  $W_{hh}$  hidden state, time  $t-1$ , Weight matrix  $W_{xh}$  for the input at time  $t$ , Weight matrix is  $W_{hy}$  for the output, Bias terms are  $b_h, b_y$ , Activation function is  $f$  for hidden state, activation function is  $g$  usewd for output. RNNs can be trained using a technique known as BPTT, which is an expansion of the traditional back propagation algorithm designed to handle the recurrent nature of the network. This approach has found application in a wide array of tasks. For instance, in Natural Language Processing (NLP), it has been instrumental in tasks such as machine translation, sentiment analysis, and named entity recognition. Moreover, RNNs have been employed in Time Series Prediction tasks, ranging from forecasting stock prices and weather patterns to energy load forecasting. Furthermore, RNNs have been pivotal in Speech Recognition endeavors, where they facilitate the conversion of spoken language into text. In the domain of Computer Vision, they have been utilized for Image Captioning tasks, generating descriptive textual summaries for images. Additionally, RNNs have made notable contributions in the realm of Music Generation, enabling the creation of new music sequences based on existing ones. Nonetheless, RNNs encounter certain challenges. Among these is the problem of vanishing and exploding gradients, which can impede the network's ability to effectively learn long-term dependencies. Another challenge is memory limitations – capturing long-term dependencies in sequences can be tricky for basic RNN architectures. Furthermore, RNNs can be slow to train, especially when dealing with long sequences. There are different types of RNNs that have been developed to address these challenges. The first type is Vanilla RNNs, which are the basic form of RNNs as described above. More advanced architectures such as LSTM have been developed to address the vanishing gradient problem, incorporating memory cells to facilitate learning long-term dependencies. Another notable architecture is the GRU, which streamlines the network by merging input gates. Despite these challenges, RNNs have been instrumental in the progress of sequence modeling, opening avenues for numerous applications across diverse fields such as natural language processing and time series analysis. Following statements have explained in figure 1.

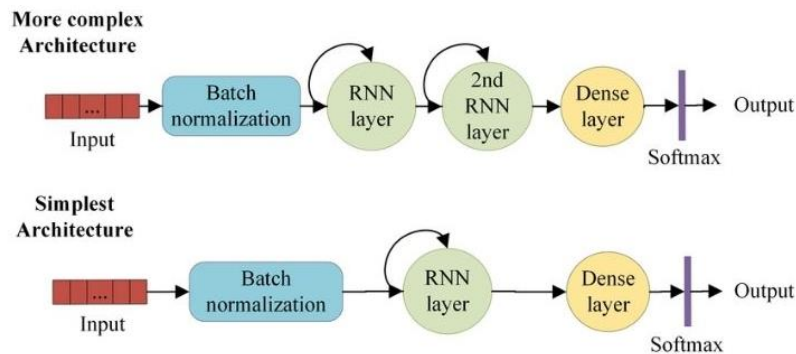


Figure 1 - Structure of RNN

### 3.2 LSTM Networks:

LSTM networks, introduced by Sepp Hochreiter and Jürgen Schmidhuber in 1997, represent a specialized form of RNNs designed to overcome the limitations of conventional RNNs in capturing long-term dependencies within sequences. They have gained widespread popularity and effectiveness in tasks involving sequential data. The



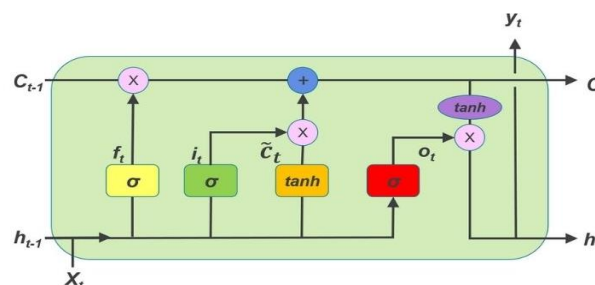
structure of LSTMs involves memory blocks that include gates to regulate the movement of data. The central concept of LSTMs is the cell state, which remains constant throughout the chain and only has minimal linear interactions. These cell states function as a conveyor belt, facilitating the transfer of information without alteration, except for a few linear interactions. This design allows LSTMs to retain information for extended periods. The LSTM cell have a main components are:

- $C_t$ : This is the line at the top of the diagram that runs vertically and allows for information to flow without interference, preserving the information.
- $f_t$ : This gate, known as the "forget gate," plays a crucial role in determining what information to retain or discard from the cell state. It takes inputs from both the previous hidden state  $h_{t-1}$  and the current input  $x_t$ , generating a value between 0 and 1 for each element in the cell state. A value of 1 indicates "keep this," while a value of 0 indicates "discard this."
- $i_t$ : The "input gate" is responsible for determining the new information to be incorporated into the cell state. This gate comprises two components: a sigmoid layer, which determines the values to be updated, a tanh layer, which generates a vector of new potential values to be added to the state.
- $o_t$ : The "output gate" merges the current input  $x_t$  and the previous hidden state  $h_{t-1}$  to decide the output based on the cell state. Its function results in a filtered version of the cell state being produced as the output. LSTM cell equations governing as follows in the table 2:

**Table 2 - LSTM Cell Equations**

Gates	Equations
Forget Gate ( $f_t$ )	$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$
Input Gate ( $i_t$ )	$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$ $\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$
Update the Cell State ( $C_t$ )	$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$
Output Gate ( $o_t$ )	$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$ $h_t = o_t * \tanh(C_t)$

Where, sigmoid function( $\sigma$ ),  $\tilde{C}$  - new candidate values,  $C_t$  - updated cell state,  $h_t$  - hidden state,  $W$  and  $b$  are weight matrices and bias vectors. Overall, LSTM networks have demonstrated their effectiveness in capturing complex relationships in sequential data, making them an essential tool in deep learning and artificial intelligence. As discussed in above statements explained in fig. 2.



**Figure 2 - Structure of LSTM**

**3.3 GRU Networks:**

GRUs represent a variant of the RNN framework designed to address the limitations of traditional RNNs, particularly in terms of their training speed and ability to grasp long-term relationships within sequential data. The

architecture of GRUs shares similarities with LSTMs, both capable of capturing long-term dependencies, but GRUs are simpler in design. GRUs consolidate the input gates of LSTMs into a single "update gate" while also incorporating a "reset gate" to control the amount of past information to forget.

Key components of GRU cell:

1. Update Gate ( $z_t$ ): This gate determines the amount of previous data to retain and the amount of new data to incorporate. It integrates the forget and input gates found in an LSTM system.

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{2}$$

2. Reset Gate ( $r_t$ ): This entrance dictates the amount of previous data to discard.

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{3}$$

3. Current Memory Content ( $\tilde{h}_t$ ): This refers to the latest candidate initiation.

$$h_t = \tanh(W \cdot [r_t \odot h_{t-1}, x_t]) \tag{4}$$

4. Update the Hidden State ( $h_t$ ): This merges a prior concealed state with memory content present.

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{5}$$

Here,  $\sigma$  sigmoid function, element-wise multiplication  $\odot$ , ( $z_t$ ) update gate, ( $r_t$ ) reset gate, ( $\tilde{h}_t$ ) current memory content,  $h_t$  hidden state,  $W_z, W_r, W$  weight matrices. Overall, Because of their efficiency, simplicity, and effectiveness in capturing dependencies, GRUs have emerged as a preferred option for tasks that involve sequential data within the realms of deep learning and artificial intelligence. Despite being simpler than LSTMs, they have proven to be powerful tools for various applications. As discussed the above statements in figure 3.

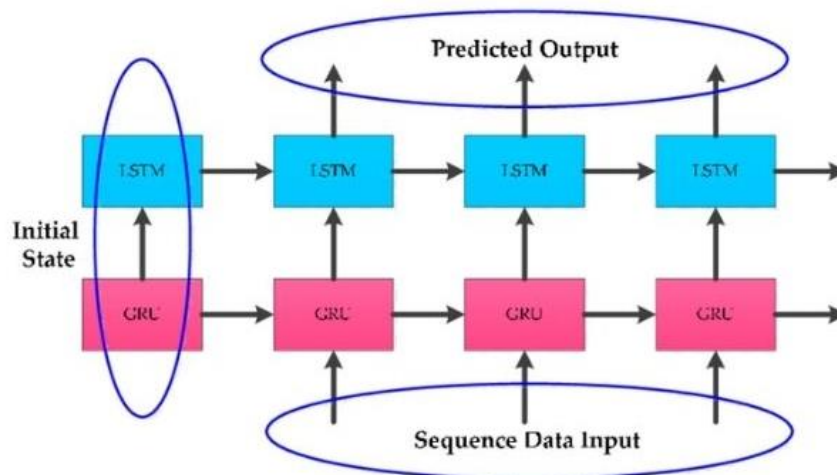


Figure 3 - Structure of GRU

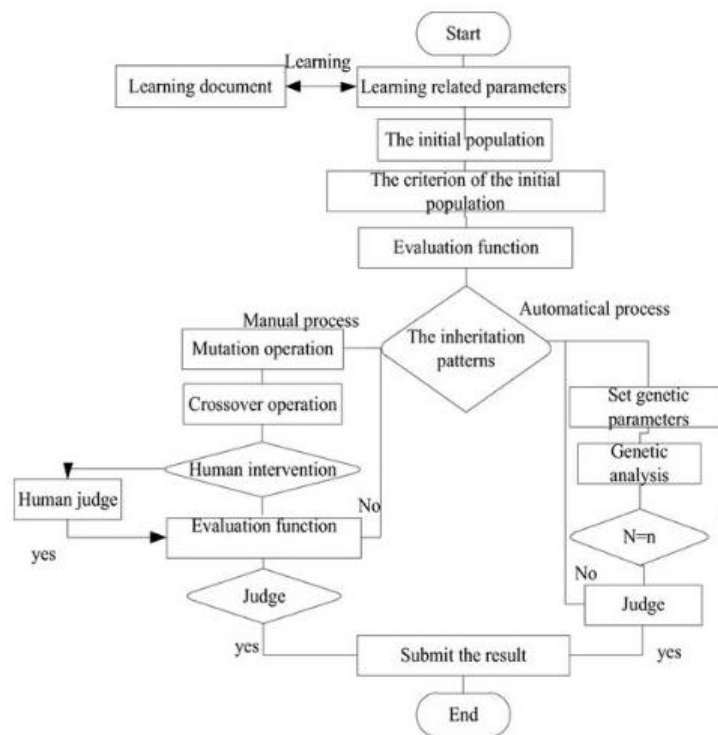
#### IV. AUTOMATIC MUSIC COMPOSITION SYSTEM (AMCS-RNN) DESIGN:

A system known as AMCS, which utilizes the RNN algorithm, is a type of computerized system that leverages the capabilities of RNNs specifically designed for sequential data, with the goal of producing musical compositions.

**4.1 RNN based on the Composition System of Music:**

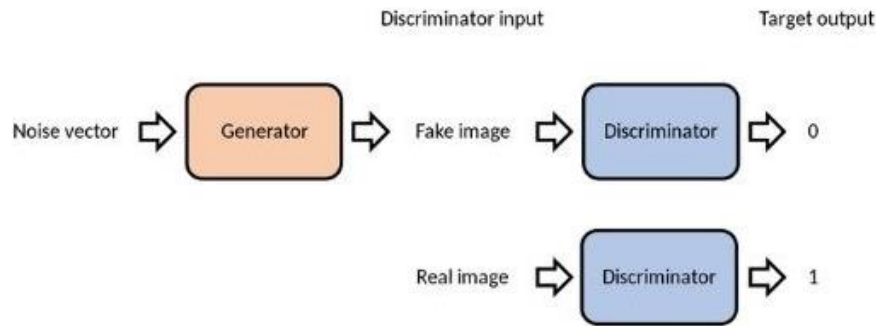
The proposed method introduces an advanced GAN (Generative Adversarial Network) model specifically designed for composing melodies based on bars. Figure 4 outlines the entire process of the system for melody composition. Initially, the pretreatment phase involves the generation of the requisite training data for the model. An algorithm is employed to differentiate the high frequencies of the melody from non-melodic elements. Melodies are then extracted from the music to serve as training data, utilizing a filter based on a shallow structural description. These extracted melodies are further segmented into individual bars using tempo information acquired from the MIDI file.

Subsequently, each bar, containing pitch, start time, duration, and intensity features for every note, is encoded into a high-dimensional matrix. These encoded bar matrices are then standardized to a consistent dimension size. Advancing to the MIDI file generation stage, new melody matrices are created through training with the enhanced GAN model using the preprocessed training data. The enhanced GAN model consists of a single generator and two discriminators. These discriminators include an RNN-based discriminator, capable of considering melody characteristics as time-series data, and a CNN-based discriminator, which analyzes the overall melody structure. During operation, the generator generates new melody matrices based on noise vectors and presents them to both discriminators for evaluation.



**Figure 4 - A Proposed Melody Composition System.**

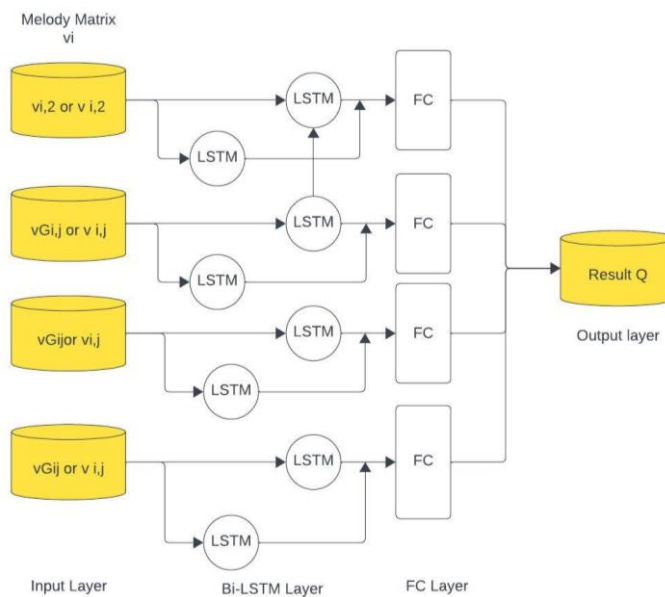
The evaluators then determine if the input melodic matrices were created by the generator or taken from preexisting MIDI files. All three components - the discriminators and the generator - are trained based on the assessments made by the evaluators. The inclusion of an RNN network, specifically LSTM, is critical for handling sequential data with time-related attributes. LSTM introduces key elements such as input gates, forget gates, output gates, and memory states which are not present in traditional RNNs. This design addresses the common problems of gradient disappearance and explosion in RNNs by incorporating these new structures. The flow of data through each gate is controlled by the sigmoid function. During processing, the input gate retrieves information from each time step's input bar and adds it to the memory state. Conversely, the forget gate selectively discards certain information from the memory state based on the input bar. As a result, the updated memory state at each time step predicts the next input bar. This predicted bar then becomes the input for the subsequent time step after sampling. Figure 5 shows a depiction of the generator's structure (G).



**Figure 5 - An Overview of Generator (G)**

By repeatedly following this process, a sequence of bars with time-related characteristics, representing melodies, can be produced. The data flow is summarized as follows: The input to the generator (G) is a group of noise vectors (N), which contains the same number of vectors as the desired amount of bars to be generated. For each melody matrix ( $v_{Gij}$ ) generated by G, it consists of an equal number of bar matrices ( $v_{Gij}$ ) as the intended amount for generation. The Bi-LSTM layer comprises a forward and backward LSTM layer. Unlike traditional models, Bi-LSTM allows for simultaneous analysis of melodies from both directions, resulting in more accurate evaluations. The inputs to the discriminator (R) include  $v_{ij}$  generated by G and  $v_{ij}$  extracted from a set of matrices (V). The forward LSTM layer processes  $v_{Gij}$  and  $v_{ij}$  in a forward direction, while the backward LSTM layer analyzes them in reverse. These layers are connected to the FC layer, which determines the final result, QR.

In this proposed enhanced GAN model, a second discriminator (C) is introduced to ensure the rationality of the produced bars. Unlike traditional models, this one incorporates two discriminators. C utilizes a CNN model with two hidden layers: a convolutional layer and an FC layer. CNNs are well-known for their effectiveness in tasks such as image recognition and classification. Our approach focuses on composing melodies based on bars rather than individual notes. Thus, each bar is encoded into a high-dimensional matrix similar to an image. Through convolutional operations, the key features of each bar are carefully extracted, allowing for a detailed analysis of its internal structure. In this setup, G represents the generator while R and C serve as discriminators. N denotes the set of noise vectors and V refers to a collection of melody matrices with various melody tracks encoded within them. Each  $v_i$  represents a melody matrix extracted from V. Figure 6 illustrates the structure of the discriminator (C).



**Figure 6 - An Overview of Discriminator (C)**

Discriminators C and R within the proposed GAN model are responsible for deriving determination results based on the correlation between each bar and the rationality of the bars, respectively. These results are utilized to train

the discriminators by maximizing the loss, as depicted in Equations (6), employing the functions,  $Loss_R()$ ,  $Loss_C$ , respectively.

$$\begin{aligned} Loss_R(R, G) &= E_{vi}[\log R(vi)] + E_N[1 - \log R(G(N))] \\ Loss_C(C, G) &= E_{vi}[\log C(vi)] + E_N[1 - \log R(G(N))] \end{aligned} \tag{6}$$

The training process for the generator, outlined in Algorithm 1, involves optimizing the loss function  $Loss_G()$  (as defined in Equation (7)). This loss function is designed to take into account the determination results provided by both discriminators, C and R, in a comprehensive manner.

$$Loss_G(R, C, G) = \frac{E_N[1 - \log(G(N))] + E_N[1 - \log C(G(N))]}{2} \tag{7}$$

Algorithm 1. Training Process for the Generator:

```

START
  Initialize generator G, discriminator R, discriminator C
  Initialize N
  For  $j \leftarrow 1$  to iterations count
    For  $i \leftarrow 1$  to batch size
       $vi$  From V
      Inform R by  $Loss_R(R, G)$ 
      Inform C by  $Loss_C(C, G)$ 
      Inform G by  $Loss_G(R, C, G)$ 
    End
  End
STOP

```

### 4.2 Extraction and Classification

Initially, each of the audio clips underwent segmentation into smaller segments. This step was essential due to the substantial variation in spectral characteristics across the entire clip, which posed challenges during analysis. The clips were segmented into frames, each consisting of 256 sample points, with an overlap of 100 sample points between consecutive frames. An observation made in various frames revealed discontinuities, which manifested as spectral leakage. To mitigate this issue, the frames were multiplied by a windowing function. In this case, the Hamming window was employed for this purpose.

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right) \tag{8}$$

An extraction in standard LSF features was performed on the segmented clips in a frame-by-frame fashion. LSF was selected for its superior quantization capability and effectiveness. the method, a audio signal represent the output of an all-pole filter  $H(z)$ , with its inverse denoted as  $R(z)$ . Here,  $R_{1..m}$  represent the predictive coefficients.

$$\begin{aligned} R(z) &= 1 + r_1 z^{-1} + \dots + r_m z^{-m} \\ R_x(z) &= R(z) + z^{-(m+1)} R(z^{-1}) \\ R_y(z) &= R(z) + z^{-(m+1)} R(z^{-1}) \end{aligned} \tag{9}$$

The LSF was drive by decomposing  $R(z)$  into  $R_x(z)$  and  $R_y(z)$ , as described below. The characteristics of different sizes were gathered, with each size corresponding to a specific range of frequencies. Afterwards, the total of coefficients for each range was calculated, allowing for the ranking of ranges from lowest to highest. This sequence of ranges provided understanding into the presence of energy across various frequencies. Additionally, statistical measures like average and standard deviation were computed for these ranges. Then, the change in rank and statistical measures between consecutive ranges was calculated to capture any variations. These change in rank and statistical measure values were used as features. The 5, 10, 15, 20 and 25 dimensions were calculated for the audio clips in a frame-by-frame manner.

For example, a 1-second clip resulted in 440 frames, resulting in a feature space of 2200 dimensions for just the 5-dimensional LSFs. However, LSF-deltaG-deltaS features of different dimensions -13, 28, 43, 58 and 73- were obtained for the LSFs ranging from 5 to 25 dimensions. These features were not affected by the length of the clip or lower dimensions. Deep learning is currently considered one of the most widely adopted and effective methods in machine learning. Its prowess has been prominently showcased across diverse domains of pattern recognition. In this context, a RNN based classifier employing LSTM was utilized. This architecture possesses a distinct advantage in preserving states compared to conventional neural networks, making it adept at handling sequences.

The AMCS-RNN framework serves as a versatile deep learning technique, adept at handling lengthy sequences. It notably tackles the vanishing gradient challenge often encountered in basic RNNs, providing an advantage, especially in complex and prolonged scenarios. At the heart of an LSTM block lies a crucial component called the cell state, which retains long-term memory. This structure incorporates three essential gates: the forget gate, input gate, and output gate. The input gate (in) assumes a critical role in generating the values required to compute the new state, as illustrated below:

$$i_n = \sigma(W_i S_{n-1} + W_i X_n), \tag{10}$$

the values required to compute the new state of the generation input gate (in) :

$$f_n = \sigma(W_f S_{n-1} + W_f X_n), \tag{11}$$

the retention or discarding of values from the previous state in the current state, regulates forget gate:

$$o_n = \tanh(W_o S_{n-1} + W_o X_n) \tag{12}$$

the values necessary for determining the next state produces output gate:

$$c_n = \tanh(W_c S_{n-1} + W_c X_n) \tag{13}$$

The computation of the cell state  $C_n$  intermediate cell state  $c_n$  is presented:

$$\begin{aligned} C_n &= (i_n * c_n) + (f_n * c_{n-1}) \\ h_n &= o_n * \tanh(C_n) \end{aligned} \tag{14}$$

Finally, the new state  $h_n$  is generated as follows:

$$h_n = o_n * \tanh(C_n) \tag{15}$$

During the study, a 100-dimensional LSTM layer was followed by three fully connected layers of sizes 100, 50, and 25. These layers used ReLU activation functions. The last layer was a 2-dimensional fully connected layer with softmax activation. At first, the training was set to 100 iterations and a 5-fold cross-validation method was used. Following experimental trials, the network architecture and its associated parameters were finalized and kept constant for subsequent evaluations. This approach ensured a consistent and stable experimental setup for the AMCS-RNN classifier.

V. EXPERIMENTAL DEMONSTRATION:

5.1 Proposed AMCS-RNN Performance

The core component of the model is a multi-layer LSTM, supplemented by a dropout layer and a time-distributed dense layer. The dropout layer is included to prevent overfitting, while the time-distributed dense layer handles the outputs at each timestep. The SoftMax classifier is chosen to accommodate the multi-class classification problem. In order to predict multi-label outputs, the SoftMax classifier is utilized. This type of classifier uses cross-entropy loss to convert raw class scores into positive numbers that sum up to one, ensuring better performance with cross-entropy loss. Figure 7 illustrates that the LSTM model contains 1904k parameters.

Layer (type)	Output Shape	Param #
embd_1 (Embedding)	(16, 64, 512)	44544
lstm_first (LSTM)	(16, 64, 256)	787456
drp_1 (Dropout)	(16, 64, 256)	0
lstm_4 (LSTM)	(16, 64, 256)	525312
dropout_4 (Dropout)	(16, 64, 256)	0
lstm_5 (LSTM)	(16, 64, 256)	525312
dropout_5 (Dropout)	(16, 64, 256)	0
time_distributed_2 (TimeDistributed)	(16, 64, 87)	22359
activation_2 (Activation)	(16, 64, 87)	0
=====		
Total params: 1904983 (7.27 MB)		
Trainable params: 1904983 (7.27 MB)		
Non-trainable params: 0 (0.00 Byte)		

Figure 7 – LSTM Model

5.1.1 Automatic Music composed by model: Below figure 8 shows LSTM model is generated with the presence of epoch number weight.

```

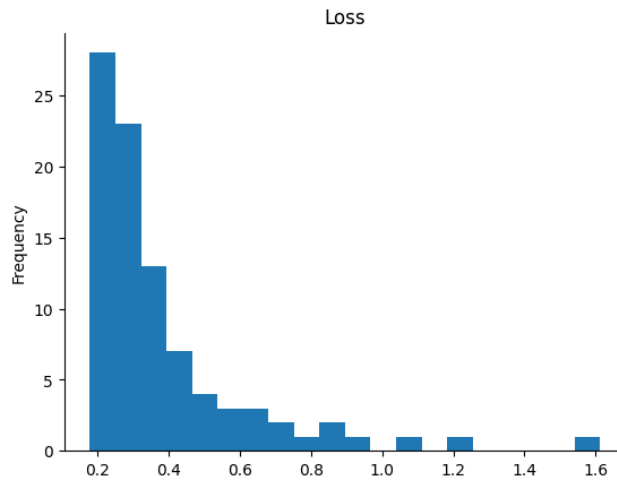
1. Which epoch number weight you want to load into the model(10, 20, 30, ..., 90). Small number will generate more errors in music: 20
2. Enter any number between 0 to 86 which will be given as initial character to model for generating sequence: 50
3. Enter the length of music sequence you want to generate. Typical number is between 300-600. Too small number will generate hardly generate any sequence: 400

MUSIC SEQUENCE GENERATED:

"F"FAF "A7"GEE|"D"FED "G"GFG|"Em"A2B "Am"ABG|"G"GFG "A7"EFG|"D"FGF "E7"D2d|
"A"efe "A7"cbC|"D"d3 "A7"d2g/2g/2|"D"f,ga faf|"F#m"edc "G"d^ed|
"D"dcd dBd|"A7"fed "Bm"dcB|"Em"AGB "A7"cbA|"D"d3 d2:|
    
```

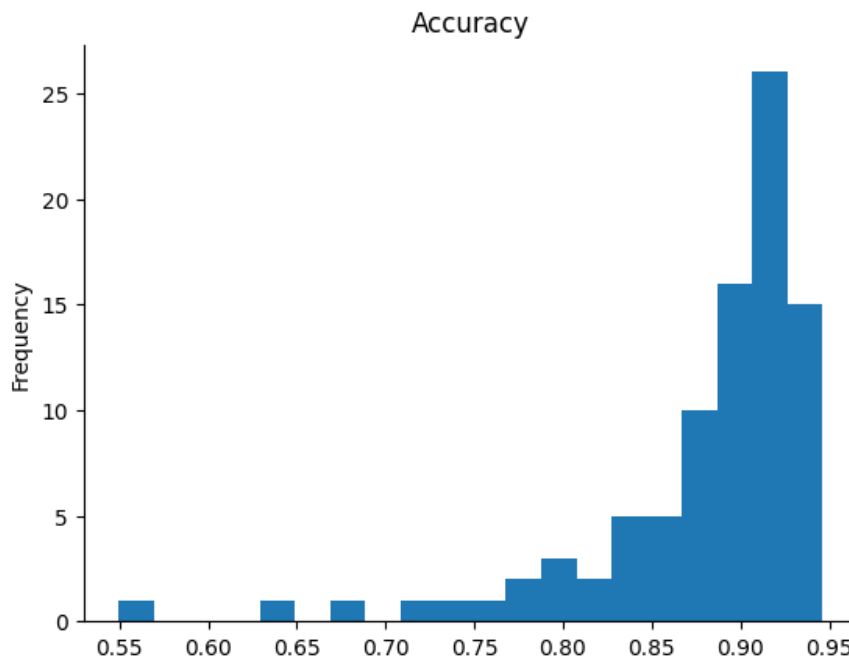
Figure 8 – Music Generation of LSTM

5.1.2 Frequency vs Loss Calculation: Prior to training the RNN, it is essential to examine the frequency distribution of musical components within the training dataset. This is necessary to ensure that the model is exposed to a diverse array of musical patterns and does not develop a preference for specific elements. The loss is computed by comparing the model's predictions during training with the actual musical elements present in dataset. By utilizing back-propagation and optimization techniques such as gradient descent, the model can gradually reduce this loss and gain a deeper understanding of the underlying patterns in the musical data. Figure 9 shows the loss vs frequency plot which depicts the minimum loss at high frequency and less frequency at high loss.



**Figure 9 – Frequency and Loss Calculation**

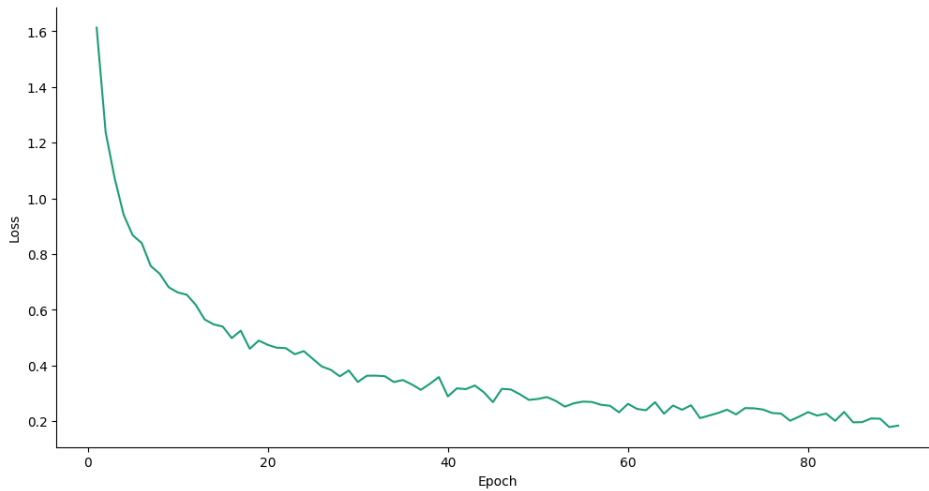
**5.1.3 Frequency vs Accuracy Calculation:** Before beginning to train the RNN, it is essential to examine the frequency of various musical components within the training dataset. This process aids in the model's comprehension of the distribution of musical patterns and enables it to generate compositions that are in line with the observed frequencies. Once trained, accuracy is measured by assessing how closely the model's generated compositions match the expected musical patterns. This metric serves as a quantitative measure of how accurately the model can replicate the frequency distribution observed in the training data. Figure 10 shows the accuracy vs frequency plot which depicts the maximum accuracy at high frequency and less frequency at low accuracy.



**Figure 10 – Frequency and Accuracy Calculation**

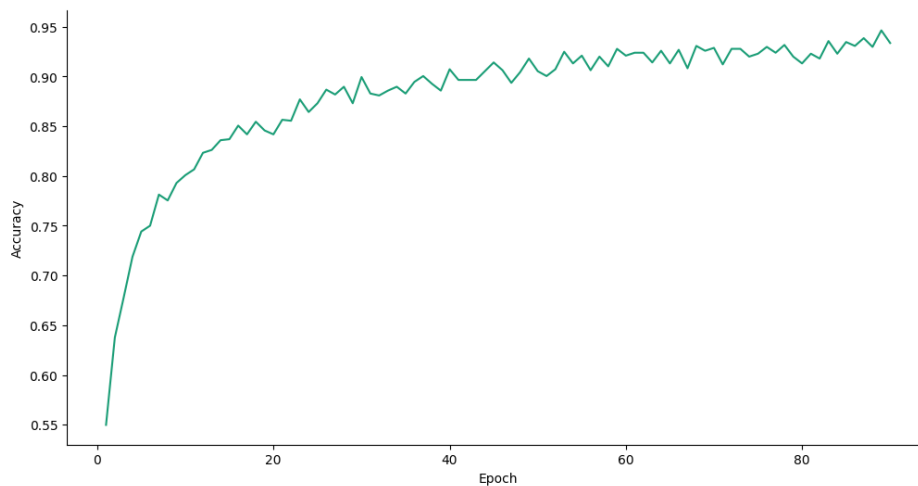
**5.1.4 Epoch vs Loss Calculation:** Loss is defined as that every epoch, the model makes predictions for the musical data in the training set, and the loss is determined by comparing these predictions to the real musical elements. The current performance of the model is measured by this loss, with higher values indicating a larger difference between predictions and actual data. The training process involves multiple epochs, during which the model fine-tunes its internal parameters to better capture patterns in the training data. The ultimate objective is for the model to accurately produce music that reflects these desired patterns. As shown in Figure 11, there is a clear decrease in training loss as the number of epochs increases, from 0.3822 at 40 epochs to a final loss of 0.1901 after 900 epochs.





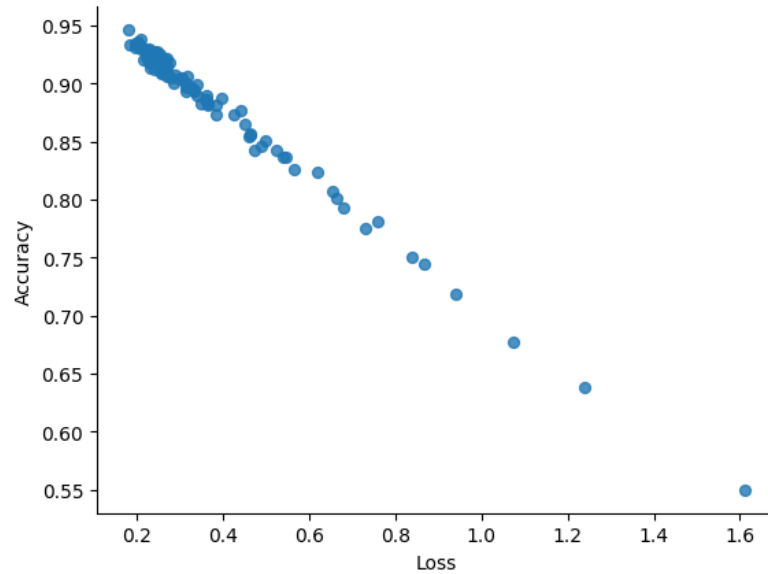
**Figure 11 – Epoch Vs Loss Calculation**

**5.1.5 Epoch vs Accuracy Calculation:** Accuracy calculation in each cycle is defined as that the model makes predictions for the musical data in the training set. After training or during evaluation, accuracy is determined by comparing the model's generated compositions with the expected musical patterns. This measure provides a numerical representation of how accurately the model can replicate the desired musical elements. The calculation of epoch is the training process involves repeating multiple cycles. With each cycle, the model adjusts its internal parameters in an effort to enhance its accuracy in creating music that corresponds to the desired patterns. Tracking accuracy over several cycles allows for evaluation of the model's effectiveness in learning and adapting to unfamiliar musical data. The model is executed for a total of 90 cycles, leading to a decrease in training loss with each successive cycle. This ultimately leads to a training accuracy rate of 95%. A visual representation of this can be seen in Figure 12, where the graph shows the relationship between epochs and accuracy. It displays the progression of training accuracy values as they correspond to the number of cycles. At 20 cycles, the training accuracy is 85% and then steadily rises from 50 cycles onward, reaching a final rate of 95%.



**Figure 12 – Epoch Vs Accuracy Calculation**

**5.1.6 Loss vs Accuracy Calculation:** The model's loss is calculated by contrasting its forecasts with the actual musical elements during both training and evaluation. This serves as a measure of the model's effectiveness, with a lower loss indicating a strong alignment between predictions and real data. On the other hand, accuracy evaluates the overall correctness of the model's predictions by comparing the number of correct predictions to the total number of predictions. A higher accuracy value shows a greater proportion of correct predictions. Figure 13 shows the loss and its corresponding accuracy of epochs. Initially, loss is high and accuracy at low epochs and then accuracy is gradually increased and loss is reduced at high epochs.



**Figure 13 – Loss Vs Accuracy Calculation**

The table 3 shows that the proposed AMCS-RNN provides high accuracy 95% than traditional LSTM accuracy 92% and low Mean Square Error 0.097.

**Table 3 – Performance Analysis**

Approaches	Accuracy	MSE
LSTM	92%	0.24
Proposed AMCS-RNN	95%	0.097

**VI. CONCLUSION:**

The use of RNNs in the AMCS has revolutionized computer-generated music. By utilizing RNNs, the system can recognize and learn complex patterns and connections within musical sequences, resulting in compositions that are highly intricate and innovative. Through extensive training with a large dataset of musical examples, the system is capable of producing compositions that not only imitate but also push the boundaries of existing musical styles. This AMCS, based on RNNs, offers a promising future for music creation as it complements human creativity and expands the possibilities of musical expression. With ongoing advancements, this technology has the potential to revolutionize our comprehension of music composition and make significant contributions to the constantly evolving realm of artistic creation.

**REFERENCES:**

- [1] Fengbin Lou, Yaling Lu, et.al, “Design of a Semantic Understanding System for Optical Staff Symbols”, Application science, vol. 13, pp. 12627, 2023, doi: 10.3390/app132312627
- [2] Shuyu Li, Sejun Jang, et.al, “Melody Extraction and Encoding Method for Generating Healthcare Music Automatically”, Electronics, vol. 8, pp. 1250, 2019, doi: 10.3390/electronics8111250
- [3] Shuyu Li, Sejun Jang, et.al, “Automatic Melody composition Using Enhanced GAN”, Mathematics, vol. 7, pp. 883, 2019, doi: 10.3390/math7100883
- [4] María Navarro-Cáceres, Wataru Hashimoto, et.al, “Sensing a Generative System to Create User-Controlled Melodies”, Sensors, vol. 18, pp. 3201, 2018, doi: 10.3390/s18103201
- [5] Miguel Civit, Javier Civit-Masot, et.al, “A systematic review of artificial intelligence-based music generation: Scope, applications, and future trends”, Expert Systems With Applications, vol. 209, pp. 118190, 2022, doi: 10.1016/j.eswa.2022.118190
- [6] Yunli Chen, Haiyang Zheng, “The application of HMM algorithm based music note feature recognition teaching in universities”, Intelligent Systems with Applications, vol. 20, pp. 200277, 2023, doi: 10.1016/j.iswa.2023.200277
- [7] María Alfaro-Contreras, Jose J. Valero-Mas, et.al, “Late multimodal fusion for image and audio music transcription”, Expert Systems With Applications, vol. 216, pp. 119491, 2023, doi: 10.1016/j.eswa.2022.119491

- [8] Vishnu S. Pendyala, Nupur Yadav, Chetan Kulkarni, et.al, "Towards building a Deep Learning based Automated Indian Classical Music Tutor for the Masses", *Systems and Soft Computing*, vol. 4, pp. 200042, 2022, doi: 10.1016/j.sasc.2022.200042
- [9] Weiwei Bao, "Music Emotion Analysis Based on Multimodal Intelligence", *Procedia Computer Science*, vol. 228, pp. 559–567, 2023, doi: 10.1016/j.procs.2023.11.064
- [10] Manuel Anglada-Tort, Peter M.C. Harrison, et.al, "Large-scale iterated singing experiments reveal oral transmission mechanisms underlying music Evolution", *Current Biology*, vol. 33, pp. 1472–1486, 2023, doi: 10.1016/j.cub.2023.02.070
- [11] Leonor Neves, Ana Isabel Correia, et.al, "Does music training enhance auditory and linguistic processing? A systematic review and meta-analysis of behavioral and brain evidence", *Neuroscience and Biobehavioral Reviews*, vol. 140, pp. 104777, 2022, doi: 10.1016/j.neubiorev.2022.104777
- [12] Mohammad Javad Mollakazemi, Dibyajyoti Biswal, et.al, "Effects of breathing pathway and musical features on the processing of music induced emotions", *Neuroscience Informatics*, vol. 3, pp. 100117, 2023, doi: 10.1016/j.neuri.2023.100117
- [13] Himadri Mukherjee, Ankita Dhar, et.al, "Music chord inversion shape identification with LSTM-RNN", *Procedia Computer Science*, vol. 167, pp. 607–615, 2020, doi: 10.1016/j.procs.2020.03.327
- [14] Mohit Dua, Rohit Yadav, et.al, "An Improved RNN-LSTM based Novel Approach for Sheet Music Generation", *Procedia Computer Science*, vol. 171, pp. 465–474, 2020, doi: 10.1016/j.procs.2020.04.049
- [15] Li Shu, "Development of key technologies for audio retrieval and vocal teaching system based on sensor networks", *Measurement: Sensors*, vol. 32, pp. 101048, 2024, doi: 10.1016/j.measen.2024.101048
- [16] Dominika Szeliga, Paweł Tarasiuk, et.al, "Musical Instrument Recognition with a Convolutional Neural Network and Staged Training", *Procedia Computer Science*, vol. 207, pp. 2493–2502, 2022, doi: 10.1016/j.procs.2022.09.307
- [17] Alexander Belden, Tima Zeng, et.al, "Improvising at rest: Differentiating jazz and classical music training with resting state functional connectivity", *NeuroImage*, vol. 207, pp. 116384, 2020, doi: 10.1016/j.neuroimage.2019.116384
- [18] Hua Zhen Lv, Junyi Luo, "Creative approaches in music teaching: Possibilities of Web 2.0 technologies", *Thinking Skills and Creativity*, vol. 40, pp. 100840, 2021, doi: 10.1016/j.tsc.2021.100840
- [19] Simon Høffding, Nanette Nielsen, et.al, "Mind surfing: Attention in musical absorption", *Cognitive Systems Research*, vol. 83, pp. 101180, 2024, doi: 10.1016/j.cogsys.2023.101180
- [20] Luca Turchet, Chan Nam Ngo, "Blockchain-based Internet of Musical Things", *Blockchain: Research and Applications*, vol. 3, pp. 100083, 2022, doi: 10.1016/j.bcr.2022.100083
- [21] Chuan-Kang Ting, Chia-Lin Wu, et.al, "A Novel Automatic Composition System Using Evolutionary Algorithm and Phrase Imitation", *IEEE Systems Journal*, vol. 11, no. 3, pp. 1284 - 1295, 2015, doi: 10.1109/JSYST.2015.2482602
- [22] DARIUSZ KANIA, PAULINA KANIA, et.al, "Trajectory of Fifths in Music Data Mining", vol. 8, 2020, doi: 10.1109/ACCESS.2021.3049266
- [23] IZARO GOIENETXEA, IÑIGO MENDIALDUA, et.al, "Statistics-Based Music Generation Approach Considering Both Rhythm and Melody Coherence", vol. 7, 2019, doi: 10.1109/ACCESS.2019.2959696
- [24] WENJUAN GONG, et.al, "A Deep Music Recommendation Method Based on Human Motion Analysis", vol. 9, pp. 26290, 2021, doi: 10.1109/ACCESS.2021.3057486
- [25] Hua Bai, "Convolutional neural network and recommendation algorithm for the new model of college music education", *Entertainment Computing*, vol. 24, pp. 100612, 2024, doi: 10.1016/j.entcom.2023.100612
- [26] Shipra Shukla, Haider Banka, "Markov-based genetic algorithm with  $\Sigma$ -greedy exploration for Indian classical music composition", *Expert Systems with Applications*, vol. 211, pp. 118561, 2023, doi: 10.1016/j.eswa.2022.118561
- [27] Qian Zhu, Achyut Shankar, Carsten Maple, "Grey wolf optimizer based deep learning mechanism for music composition with data analysis", Vol. 153, pp. 111294, 2024, doi: 10.1016/j.asoc.2024.111294