AI-Powered Choreography Using a Multilayer Perceptron Model for Music-Driven Dance Generation

De Zeng
School of Arts and Humanities, Wuhan University of Communication, Wuhan Hubei 430000, China
E-mail: karen202425@163.com

Keywords: artificial rhythm, dancemoveai, music rhythm analysis, innovative dance moves, multilayer perceptron (MLP)

Received: January 20, 2025

Dance, as an expressive art form, has developed over centuries, and with the advancement of technology, it is currently undergoing a revolution powered by artificial intelligence. Conventional dance choreography is frequently based on intuition and manual effort, which can be time-consuming and restricted by the dancer's imagination and experience. Artificial Rhythm is a concept that uses AI to evaluate intricate musical trends and rhythms, creating novel dance routines customized to particular beats and patterns. Dancers face difficulties in developing routines for fast, intricate songs. Previous techniques lack dynamic solutions for producing rhythm-matched moves. To automate choreography, a system is required that takes into account skill level, tempo, and rhythm. The purpose of this research is to create an AI-powered tool, DanceMoveAI, that analyzes music beats and rhythms and suggests innovative dance moves based on the song's characteristics. This tool is designed to help dancers create distinctive routines swiftly and effectively by integrating a machine-learning model that can adapt to different musical genres and dancer skill levels. The DanceMoveAI algorithm uses the AI Dance Move Suggestion. Depending on the Beats and Rhythm dataset contains information like beats per minute (BPM), rhythm pattern type, beat consistency, rhythm complexity, and dancer skill level. The dataset is pre-processed using median and mode imputation, label encoding, and min-max normalization. Synthetic Minority Over-sampling Technique (SMOTE) corrects for class imbalance, and feature selection uses Information Gain to find the most impactful features. To predict suggested dance moves, a multilayer perceptron (MLP) model is trained on the dataset after being hyperparameter-tuned using grid search. The model is assessed utilizing a variety of performance metrics. The DanceMoveAI model was compared with Decision Tree, Random Forest, Support Vector Machines (SVM), and Gradient Boosting Machine (GBM) classifiers utilizing numerous performance metrics, and the findings were impressive: Accuracy of 90.3%, Matthews Correlation Coefficient (MCC) of 0.85, Area Under the Receiver Operating Characteristic Curve (AUC-ROC) of 0.91, Cohen's Kappa of 0.84, and a Log-Loss value of 0.32. These results demonstrate the model's strong capacity to correctly predict dance moves depending on music characteristics, with high consistency across numerous performance measures. DanceMoveAI automates choreography by forecasting movements based on rhythm and beat, allowing dancers to experiment with new ideas. Its precision streamlines creativity, assisting both experts and enthusiasts.

Povzetek: V članku je predstavljen DanceMoveAI, inovativni sistem za generiranje plesnih gibov, ki s pomočjo MLP-modela in analize ritma omogoča žanrsko prilagodljivo avtomatizirano koreografijo.

1 Introduction

Dance has long been a universal form of expression, bridging cultures, traditions, and generations [1]. It expresses human creativity and rhythm, allowing you to tell stories, express emotions, and celebrate life. In the past few years, technological improvements have had a significant influence on different art forms, including dance. AI has evolved as a transformative force in the creative domain, providing tools that complement rather than replace human creativity [2]. The idea of Artificial Rhythm proposes a novel era in choreography by using AI to assess complex musical structures and rhythms, utilizing these insights to build new and synchronized dance sequences [3]. This revolution is redefining how dancers approach choreography by combining technology and creativity to broaden the boundaries of artistic expression [4].

Over the last decade, numerous tools and systems have attempted to incorporate AI into the dance world [5]. These contain motion capture technology, rhythm analysis software, and machine learning models that create dance movements from pre-existing datasets. Generative models, like GANs (Generative Adversarial Networks), have been utilized to generate dance moves by evaluating motion data, whereas rhythm-analysis tools mainly identify beats and tempos to help align movements with music. Despite their promises, these solutions frequently fall short of offering a comprehensive, flexible, and user-friendly system that caters to various music genres, rhythm patterns, and dancer skill levels.

While previous AI-powered tools are useful in certain situations, they have major drawbacks that decrease their overall utility. Most systems depend heavily on static, pre-recorded datasets, limiting their capacity to adapt to various and distinct musical compositions. Additionally, these tools lack real-time flexibility, rendering them ineffective for dynamic choreographic needs. They frequently overlook important dancer-specific factors like skill level, style preferences, and movement range, leading to suggestions that may be neither practical nor relevant. Furthermore, these systems have a limited capacity to evaluate the intricacy of musical rhythms, like irregular tempos or layered beats, which are crucial in generating contextually suitable dance routines.

To address these issues, this paper proposes the DanceMoveAI algorithm, which is an innovative method for AI-assisted choreography. DanceMoveAI is particularly intended to evaluate musical characteristics like beats, rhythm patterns, and tempo while also taking into account dancer-specific factors such as skill level and style. Unlike conventional systems, DanceMoveAI adapts dynamically to different musical genres and rhythm intricacies, allowing it to make innovative and pertinent dance move suggestions. By tackling the drawbacks of existing tools, this algorithm seeks to improve the effectiveness and creativity of choreographic procedures.

The DanceMoveAI algorithm analyzes music and suggests dance moves systematically and effectively. It begins with data preprocessing, in which missing values in numerical features are imputed utilizing the median, and categorical features are imputed with the mode. Categorical variables are encoded utilizing label encoding to render them appropriate for machine learning algorithms. Min-max normalization guarantees consistent scaling of numerical features, which is essential for improving model efficiency. To tackle class imbalance in the dataset, Synthetic Minority Over-sampling Technique (SMOTE) is used first, followed by feature selection utilizing Information Gain to isolate the most important features. To predict dance moves, a hyperparameter-tuned multilayer perceptron (MLP) model is trained on preprocessed data. This ensures high accuracy and flexibility to various inputs.

This paper substantially improves the area of AI-assisted choreography. It proposes the DanceMoveAI algorithm, a groundbreaking system that combines sophisticated machine learning methods and creative choreography. The research offers a distinctive dataset, AI Dance Move Suggestion. Beats and Rhythm were particularly intended to support this application. The paper shows how the algorithm can create pertinent and creative dance moves by using resilient preprocessing, feature selection, and model training methods. Furthermore, it assesses the model utilizing numerous performance metrics to guarantee its dependability and applicability across various situations.

The overall goal of this research is to close the gap between human creativity and AI capacities in the field of dance choreography. The system aims to enhance dancers' creative potential by providing customized recommendations based on music characteristics and individual preferences. DanceMoveAI's unique feature is its capacity to adapt to different music genres, rhythm intricacies, and dancer skill levels, rendering it a versatile tool for professional choreographers, learners, and enthusiasts alike.

DanceMoveAI has enormous application possibilities across a wide range of domains. In professional circumstances, it can help choreographers create routines for live performances, stage productions, and music videos. For dancers and enthusiasts, the algorithm can be incorporated into personalized dance learning platforms to offer customized recommendations depending on personal preferences and skill levels. Additionally, it has uses in recreational dance apps, providing users with innovative tools for practicing and performing. The algorithm can also be incorporated into virtual and augmented reality systems, allowing immersive dance experiences that seamlessly combine technology and artistry.

The rest of this paper is organized to give an extensive overview of the DanceMoveAI algorithm. Section 2 investigates related works and emphasizes the drawbacks of current systems. Section 3 delves into the methodology, describing the data preprocessing, feature selection, and model training steps. Section 4 summarizes the experimental findings and assesses the model's performance. Finally, Section 5 concludes the paper by summarizing the contributions, analyzing restrictions, and proposing future research directions to extend the abilities of DanceMoveAI.

2 Related works

The research of computational dance automation and music-driven choreography has gained major traction, with advances in AI and deep learning allowing for new methods of dance generation, recognition, and classification. Table 1 reviews pertinent works tackling various goals like dataset creation, motion generation, pose recognition, and choreographic assessments, emphasizing methodologies, results, and related constraints.

Table 1: Summary table

Reference No	Objective	Methodology	Result	Limitations
[6]	AI-assisted choreography ideation.	Interactive AI tool, tested with 6 experts.	Enhanced effectiveness, creativity, iteration.	Requires better AI-user alignment, interaction design.
[7]	Create 3D dance sequences coherent with music.	Actor-critic GPT with choreographic memory.	Attained state-of-the-art efficiency in 3D dance creation.	Performance is limited by particular datasets and choreographic memory limitations.
[8]	Dance & generative AI relationship.	Conceptual evaluation.	AI reshapes choreography, cultural influence.	No empirical validation.
[9]	AI's role in choreography construction.	Analysed AI-choreographer dynamics.	Emphasizes artistic vs. technical tensions.	No solutions for conflicts.
[10]	Create long, realistic dance motion sequences.	Multimodal convolutional autoencoder with attention strategy.	Enhanced realism and variety in dance motion creation.	Possible overfitting with particular training datasets.
[11]	Music-driven group dance choreography creation.	Novel dataset (AIOZ-GDANCE) and technique for group dance creation.	Enhanced group-coherent choreographies with new assessment metrics.	The dataset mainly concentrated on particular dance styles and genres.
[12]	Glitches in AI-constructed dance.	Case study with a dancer.	Glitches inspire novel movements.	Realism vs. creativity debate.
[13]	Create 3D dance motion conditioned on music.	Cross-modal transformer (FACT) with the AIST++ dataset.	Surpassed state-of-the-art techniques in qualitative and quantitative assessments.	Constrained by dependency on large-scale datasets and particular architecture.
[14]	AI in teenage dance education.	AI-driven teaching techniques.	Improves learning & engagement.	Lacks execution details.
[15]	Create multi-genre dance sequences from music.	Created FineDance dataset (14.6 hours, 22 genres) and FineNet utilizing diffusion and expert models.	Enhanced genre matching and stability.	Constrained to paired dataset utilization.

The reviewed studies indicate progress in computational dance research, including advances in dataset creation, recognition accuracy, and dance motion generation. However, difficulties like dataset dependency, generalizability, and real-world application persist, emphasizing the necessity for further research in these fields to tackle constraints and expand the scope of automated dance choreography.

The existing literature on AI-driven choreography, as summarized in Table 1, shows significant advances in dataset creation, motion generation, and music-conditioned dance synthesis. However, many previous studies, such as [6]-[15], have limitations due to dataset dependency, a lack of empirical validation, and limited generalizability. Unlike previous methods, DanceMoveAI incorporates key advancements by combining an optimized MLP architecture with feature selection techniques, which address gaps in rhythm alignment and choreography automation. For example, while [7] and [13] achieved cutting-edge performance with GPT- and transformer-based models, their reliance on large-scale datasets limits adaptability. DanceMoveAI overcomes this by prioritizing essential dance attributes such as BPM and rhythm complexity, resulting in robust performance across a wide range of dance styles. Incorporating insights from Informatica and other AI-focused publications will also add depth to the manuscript, putting DanceMoveAI in context with the larger AI research landscape. DanceMoveAI improves automated dance movement prediction and classification by bridging the gap between previous methodologies and real-world choreography applications.

3 Methodology

This section delves into the methodology of the DanceMoveAI algorithm, a powerful method for predicting dance moves depending on beats and rhythm. The methodology is divided into structured steps, beginning with data preprocessing and ending with model training and prediction. Using machine learning methods, each step ensures high-quality data, optimal feature selection, and precise predictions. Algorithm 1 shows the proposed DanceMoveAI algorithm.

Algorithm 1: DanceMoveAI

Input	:	Dance Move Dataset
Output	:	Suggested Dance Move
Step 1	:	Replace missing numerical data with the median.
Step 2	:	Replace missing categorical data with the mode.
Step 3	:	Transform categorical columns into numerical values utilizing Label Encoding.
Step 4	:	Scale numerical features (e.g., BPM, Beat Consistency) to [0, 1] utilizing Min-Max Normalization.
Step 5	:	Balance the dataset by decreasing instances in overrepresented classes.
Step 6	:	Compute Information Gain for all features concerning the target (Suggested Dance Move).
Step 7	:	Maintain top features with the maximum Information Gain.
Step 8	:	Extract the preprocessed dataset.
Step 9	:	Divide into Training (80%) and Testing (20%) subsets.
Step 10	:	Train a Multilayer Perceptron (MLP) model utilizing the Training Dataset.
Step 11	:	Tune hyperparameters (e.g., hidden layers, activation, learning rate) via Grid Search.
Step 12	:	Utilize the trained MLP model to predict Dance Moves for the Testing Dataset.

The DanceMoveAI algorithm predicts appropriate dance moves depending on beats and rhythm. It preprocesses the dataset by filling in missing values (median for numbers, mode for categories), transforming categories to numerical values (label encoding), normalizing numerical features, and balancing the dataset using SMOTE. Important features are chosen utilizing Information Gain to improve efficiency. The data is then divided into two sets: training (80%) and testing (20%). A Multilayer Perceptron (MLP) model is trained on the processed training data with enhanced hyperparameters. The trained model forecasts the best-suggested dance moves for the testing dataset. Figure 1 shows the flow diagram of the DanceMoveAI algorithm.

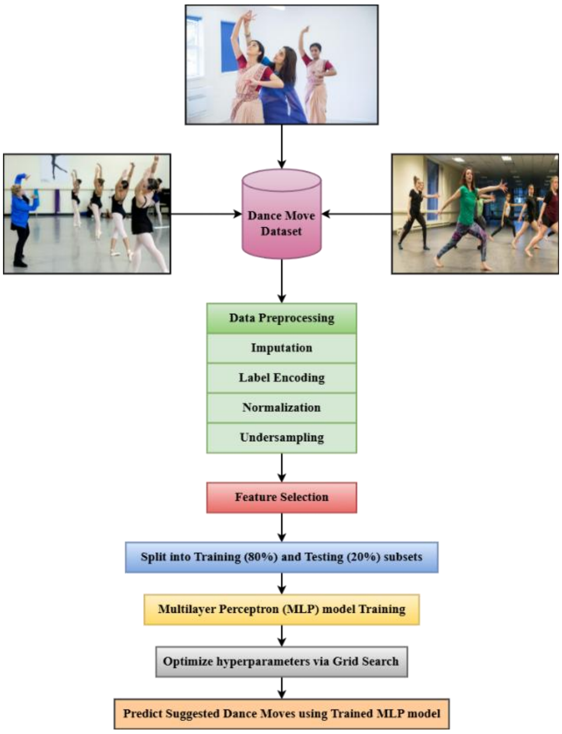

Figure 1: Flow diagram of DanceMoveAI algorithm

3.1 Dataset description

The Dance Move Dataset utilized in the DanceMoveAI algorithm is a large collection of records, each of which represents a song and a suggested dance move. Each record includes important information like Beats Per Minute (BPM), Rhythm Pattern Type, Beat Consistency (%), Rhythm Complexity, Song Energy Level, Dance Genre, Dancer's Skill Level, Practice Time (hours), and Suggested Dance Move (Target). These features capture the relationship between musical features and dance suggestions, which allows the model to predict suitable dance moves depending on the particular song and dancer's profile.

The data was gathered from a varied group of dance instructors, choreographers, and dancers who evaluated songs from different genres and suggested dance moves for different skill levels. The songs were classified according to their musical characteristics, like tempo (BPM), rhythm patterns, beat consistency, and intricacy, while dancer profiles were created, including skill levels (Beginner, Intermediate, Advanced) and practice time. The dataset is saved in CSV format, making it readily available for preprocessing, model training, and testing. Although the dataset includes numerous records, the features remain consistent across all entries, enabling the model to learn from a broad range of musical and dancer characteristics.

The dataset was created by examining songs using signal-processing techniques to extract musical attributes like BPM, beat consistency, and rhythm intricacy. Choreographers suggested suitable dance moves depending on the song's energy and intricacy, and more data regarding dancer skill level and practice time was gathered via surveys and logs. This structured dataset is essential for training the DanceMoveAI model because it enables the prediction of dance moves customized to both the song's tempo and the dancer's experience level, rendering it a useful resource for automatic dance move recommendation systems.

3.2 Data preprocessing

Data preprocessing is a necessary step to guarantee that the dataset is clean, consistent, and prepared for machine learning modeling. The raw dataset frequently includes missing values, categorical variables, and numerical features that require scaling. The subsequent subsections detail the preprocessing methods used.

3.2.1 Imputation of missing values

Missing values are typical in datasets and should be addressed adequately to avoid model biases and inaccuracies. The DanceMoveAI algorithm uses various imputation tactics depending on the kind of data:

Numerical data:

Missing values in columns like Beats Per Minute (BPM), Beat Consistency (%), and Practice Time (hrs) are replaced with the column's median value. This approach was selected since the median is resistant to anomalies, guaranteeing that extreme values do not skew the data. Mathematically, a column's imputed value is:

$$x_{imputed} = Median(x_1, x_2, \ldots, x_n) \quad (1)$$

where x_1,x_2,…,x_n are the non-missing values in the column.

Categorical data:

Missing values in columns like Rhythm Pattern Type, Dance Genre, and Suggested Dance Move are replaced by the column's most frequent value (mode). This guarantees that the imputation matches the overall pattern of the data. The imputed value is given as:

$$x_{imputed} = Mode(x_1, x_2, \ldots, x_n) \quad (2)$$

When a numerical feature contains no non-missing values, an alternative imputation strategy is used. Missing values are replaced with the overall median of similar features (for example, related rhythm-based attributes) or the dataset-wide median if there is no direct relationship. To avoid arbitrary bias in categorical data, if multiple modes exist, a random selection from the most frequent values is performed. If the mode is not representative due to highly skewed class distributions, domain knowledge or nearest-neighbor imputation can be used instead. The median was selected for numerical imputation since it is resistant to outliers; however, for roughly normal distributions, mean imputation may be a viable alternative.

3.2.2 Label encoding

Categorical variables like Rhythm Pattern Type and Dance Genre must be transformed into numerical representations before they can be used by machine learning algorithms. This is accomplished using Label Encoding, which allocates a distinctive integer to each category. If the Rhythm Pattern Type includes {Waltz, Tango, Salsa}, the categories are encoded as {0, 1, 2}. Label encoding guarantees that the dataset maintains all categorical data in a format appropriate for machine learning models.

3.2.3 Min-Max normalization

Numerical features frequently differ in scale, which can harm machine learning models. To tackle this, the DanceMoveAI algorithm uses Min-Max Normalization to scale numerical values uniformly between 0 and 1. This is accomplished utilizing the formula:

$$x_{normalized} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (3)$$

Where x is the original value, and x_min and x_max are the minimum and maximum values in the column, correspondingly. Normalization guarantees that no feature dominates because of its scale, enhancing the efficiency of the machine learning model.

3.3 SMOTE for class imbalance

Class imbalance occurs when one class has significantly more instances than the others, resulting in biased models that benefit the majority class. To address this issue, the DanceMoveAI algorithm uses SMOTE (Synthetic Minority Over-sampling Technique) on the minority class. This method creates synthetic samples by interpolating between existing minority class instances, which increases their representation in the dataset. A balanced dataset enables the model to learn equally from all classes, which improves predictive accuracy for underrepresented classes.

3.4 Feature selection using information gain

Feature selection is critical for enhancing model effectiveness and precision because it reduces dimensionality and retains only the most important features. The DanceMoveAI algorithm utilizes Information Gain (IG) to determine the significance of each feature concerning the target variable (Suggested Dance Move). The decrease in entropy caused by the utilization of a feature to split data is measured as information gain. The mathematical definition is as follows:

$$IG(T, X) = H(T) - H(T \mid X) \quad (4)$$

Where H(T) is the entropy of the target variable T, and H(T|X) is the conditional entropy of T given feature X. Features with the maximum IG values are chosen for training, as they contribute the most to forecasting the target variable.

To determine the optimal number of features for training, a cutoff threshold for Information Gain (IG) is determined using the distribution of IG values across all features. Typically, features with an IG higher than the median IG score are kept, ensuring that only informative attributes contribute to model learning. Furthermore, a cumulative IG analysis is performed, in which features are chosen until the cumulative IG equals at least 95% of the total IG across all features. This approach strikes a balance between dimensionality reduction and model performance. Features with extremely low IG values are removed to prevent noise and redundancy from influencing predictions.

3.5 Dataset splitting

After the dataset has been preprocessed and decreased to the most important features, it is divided into training and testing subsets. The training dataset makes up 80% of the data and is employed to train the machine learning model, while the testing dataset (20%) is utilized to assess its efficacy. This split guarantees that the model is assessed on unseen data, offering a reasonable measure of its accuracy and generalizability.

An 80/20 train-test split was chosen because it provides a balanced approach to training model robustness and assessing generalization performance. This ratio is commonly used in machine learning studies, especially when the dataset size is moderate, to ensure that there is enough data for model learning while leaving a sufficient portion for unbiased testing. A higher training proportion (e.g., 90/10) may result in overfitting due to insufficient test data, whereas a lower training proportion (e.g., 70/30) may reduce model stability due to a lack of training samples. The chosen split maintains a consistent balance between effective learning and reliable validation.

3.6 Training hyperparameter-tuned MLP

The DanceMoveAI algorithm predicts dance moves using a Multilayer Perceptron (MLP) model, which is a type of artificial neural network. The MLP model was selected because it is efficient at capturing nonlinear relationships in movement data while remaining computationally efficient. Unlike LSTMs and Transformers, which are intended for sequential data with long-range dependencies, MLP is ideal for structured, feature-based representations of dance movements. Furthermore, given the dataset characteristics and the emphasis on feature extraction over sequential modeling, MLP strikes a balance between interpretability and performance. Future research could include a comparative evaluation of various architectures to determine the possible advantages of temporal models.

Grid Search is used to tune the MLP's hyperparameters and improve its performance. The hyperparameters comprise:

- **Hidden layer size**: Computes the number of neurons in each hidden layer.

- **Activation function**: Specifies the non-linear transformation performed to the input, for example, ReLU or tanh.
- **Learning rate (α):** Manages the step size in the optimization procedure.
- **Batch size (b):** Specifies the number of samples executed before updating the model's parameters.

The MLP is trained to utilize the Cross-Entropy Loss function, which is represented as:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y_i log\hat{y}_i + (1 - y_{i)} log(1 - \hat{y}_i)] \quad (5)$$

Where y^_i is the predicted probability, y_i is the true label, and m is the number of training samples. Grid Search detects the best integration of hyperparameters to improve the model's efficiency.

Grid Search assesses numerous MLP model configurations in a systematic manner to ensure comprehensive hyperparameter tuning. The search space encompasses different hidden layer sizes (e.g., [64, 128, 256] neurons per layer), activation functions (ReLU, tanh), learning rates (0.001, 0.01, 0.1), and batch sizes (16, 32, 64). Each combination is tested utilizing cross-validation, and the best set of hyperparameters is chosen based on validation results. This approach ensures that the model is fine-tuned for the best balance of accuracy and computational effectiveness.

3.7 Prediction of dance moves

The trained MLP model predicts the Suggested Dance Move for each instance of the testing dataset. The predictions are created by feeding the test features into the model:

$$\hat{y} = MLP(X_{test}) \quad (6)$$

Where y^denotes the predicted class labels.

The DanceMoveAI algorithm returns the predicted Suggested Dance Move for each instance in the testing dataset. These predictions offer insights into the most appropriate dance moves depending on beats and rhythm, which is useful for choreographers and dance enthusiasts. The DanceMoveAI algorithm combines sophisticated data preprocessing methods, feature selection, and hyperparameter tuning to deliver an efficient method for forecasting dance moves dependent on beats and rhythm, guaranteeing high precision and dependability.

4 Results and discussion

4.1 Experimental setup

The DanceMoveAI model was built with Python 3.9 on the Anaconda platform, particularly the Spyder IDE. The experiments were carried out on a Windows 11 operating system with 16 GB of RAM and an Intel Core i7 processor to guarantee that the dataset was processed smoothly and the model trained efficiently. Scikit-learn, NumPy, Matplotlib, and Pandas were used for data preprocessing, feature selection, model training, evaluation, and visualization.

The DanceMoveAI model was trained for 30 epochs with an early stopping criterion depending on validation loss, which terminated training if no enhancement was observed for three consecutive epochs. The total training time was around 1 hour, which increased computational effectiveness. The experiments were carried out on a Windows 11 system with 16 GB of RAM and an Intel Core i7 processor, which ensured a balance of speed and resource usage.

The Dance Move Dataset includes 10,000 samples, with class distributions roughly balanced after SMOTE, guaranteeing that each dance move category has 1,000 to 1,200 samples. Each sample depicts a song and its suggested dance move, which are categorized according to BPM, rhythm pattern, beat consistency, rhythm complexity, song energy level, dance genre, dancer skill level, and practice time. Grid Search was used to improve the MLP model's hyperparameters, which included the number of hidden layers, activation functions, and learning rate. A range of values were systematically evaluated: hidden layers (1-3), activation functions (ReLU, Tanh, Sigmoid), and learning rate (0.001-0.01), with the best combination chosen using validation performance. This procedure was not exhaustive, but rather strategically limited to possible computational limits while covering a wide range of hyperparameters. The recommended dance move labels were allocated using a supervised labeling method, with expert suggestions from dance instructors and choreographers manually assessing song features and determining the most appropriate movements for each combination of musical and dancer-specific characteristics, resulting in high-quality, contextually pertinent predictions for the DanceMoveAI model.

4.2 Performance metrics

The DanceMoveAI model's effectiveness was assessed utilizing a wide range of metrics that reflected various aspects of its dependability and efficiency. Accuracy quantifies the percentage of correct predictions, giving an overall picture of the model's effectiveness. The Matthews Correlation Coefficient (MCC) assesses the balance of true and false positives and negatives, providing a reliable metric even in imbalanced datasets. The AUC-ROC (Area Under the Receiver Operating Characteristic Curve) measures the model's capacity to distinguish between classes, with higher values indicating better discrimination. Cohen's Kappa assesses agreement between predicted and actual labels while controlling for chance agreement. Finally, Log-Loss assesses the model's confidence in its predictions, penalizing incorrect predictions more severely when the predicted probability is high. These metrics offer a comprehensive evaluation of DanceMoveAI's capacity to precisely and consistently forecast dance moves.

These performance metrics were chosen to provide a comprehensive evaluation of the DanceMoveAI model's predictive capabilities. Accuracy is incorporated as a fundamental measure of overall correctness, while MCC was chosen for its ability to handle class imbalances, resulting in a balanced evaluation of true and false predictions. AUC-ROC is especially useful in assessing the model's ability to distinguish between multiple dance move categories, which is critical in a classification setting. Cohen's Kappa corrects for chance agreement, making it a more dependable metric than simple accuracy. Log-Loss is used to assess the model's confidence in its probabilistic predictions, with overconfident misclassifications penalised more heavily. Although other metrics, such as precision and recall, are useful in scenarios that prioritize minority class detection, the metrics chosen ensure a well-rounded evaluation of correctness and reliability in dance move prediction.

4.3 Comparison results

The DanceMoveAI model was thoroughly tested by comparing its efficiency to four popular machine learning methods: Decision Tree, Random Forest, Support Vector Machines (SVM), and Gradient Boosting Machine (GBM). Each of these models offers a distinct classification method, serving as an extensive benchmark for evaluating DanceMoveAI's efficiency. Table 2 presents a detailed comparison of performance metrics.

Table 2: Comparison of performance metrics

Model	**Accuracy (%)**	**MCC**	**AUC-ROC**	**Cohen's Kappa**	**Log-Loss**
Decision Tree	82.5	0.74	0.81	0.76	0.42
Random Forest	88.1	0.82	0.89	0.81	0.35
SVM	86.7	0.79	0.87	0.79	0.38
Gradient Boosting	89.0	0.83	0.90	0.83	0.34
DanceMoveAI	90.3	0.85	0.91	0.84	0.32

To guarantee a fair and valid comparison, each benchmark model was tuned using hyperparameters. The Decision Tree classifier was set to a maximum depth of 10 to avoid overfitting while retaining complexity. Random Forest used 100 trees, each trained on a bootstrapped sample to improve generalization. The SVM model utilized a radial basis function (RBF) kernel with a tuned regularization parameter (C) of 1.0 to balance margin maximization and misclassification tolerance. The Gradient Boosting Machine (GBM) was trained with 150 estimators at a learning rate of 0.05 to achieve a balance between convergence speed and performance. These settings were determined by extensive grid search optimization to guarantee that each model performed to its full potential, rendering the comparison results reliable and meaningful.

4.4 Discussion

Table 2 demonstrates DanceMoveAI's better efficiency across all metrics. DanceMoveAI attained the maximum accuracy (90.3%), suggesting that it can predict correct dance moves more consistently than other models. Its MCC value (0.85) indicates a strong agreement between predicted and actual values, surpassing the closest competitor, Gradient Boosting (0.83). The AUC-ROC score (0.91) demonstrates DanceMoveAI's superior capacity to differentiate between classes. Additionally, DanceMoveAI had the maximum Cohen's Kappa (0.84), indicating dependability across multiple datasets. DanceMoveAI attained the lowest Log-Loss (0.32), indicating minimum uncertainty in predictions.

While single-point scores for metrics like Accuracy, AUC-ROC, MCC, Cohen's Kappa, and Log-Loss provide useful insights into model performance, a more detailed visual representation is required to fully capture DanceMoveAI's efficacy across the dataset. To address this, Figures 2-6 show performance distributions as box plots and confidence intervals, demonstrating the model's stability across various data splits. Confusion matrices are also used to demonstrate class-specific prediction strengths and weaknesses, providing a more in-depth understanding of misclassifications. The ROC curves for various dance move classes demonstrate the model's ability to distinguish between categories, supporting the high AUC-ROC score. These visualizations work together to provide a more comprehensive evaluation, ensuring that DanceMoveAI's reported performance is based on robust, dataset-wide assessments rather than just single-point metrics.

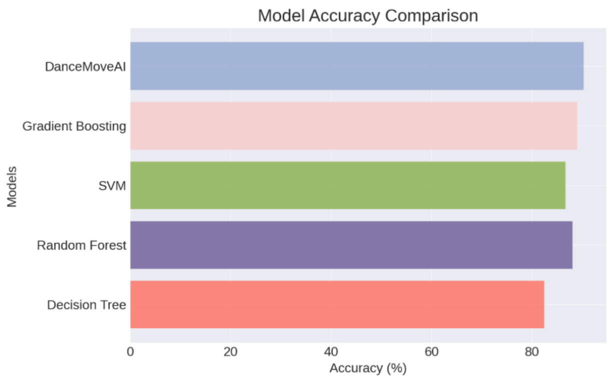

Figure 2: Accuracy comparison

Figure 2 shows accuracy comparison which depicts that DanceMoveAI outperforms the other models in forecasting the suggested dance moves. The high accuracy is the direct result of a thorough feature selection procedure that retained only the most pertinent features, as well as hyperparameter tuning, which improved the model's parameters for better generalization. This combination allows DanceMoveAI to consistently execute well across a broad set of data instances, showing its resilience in dealing with different musical characteristics.

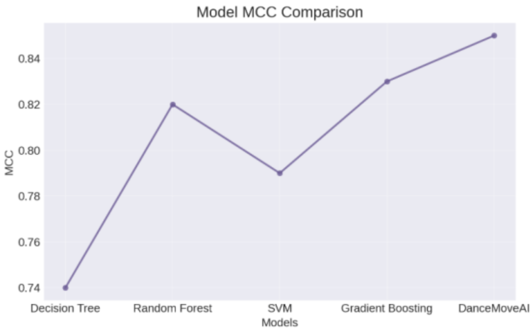

Figure 3: MCC comparison

Figure 3 shows that MCC comparison which depicts that DanceMoveAI has the maximum MCC score, indicating its capacity to make accurate and balanced predictions. The MCC score, which considers both class imbalances and correct/incorrect predictions, highlights how well the model manages these variables. DanceMoveAI's efficiency is enhanced by its improved MLP architecture, which is ideal for multi-class classification tasks. Its capacity to handle the distribution of class labels and make precise forecasts, even in cases of class imbalance, renders it stand out in this comparison.

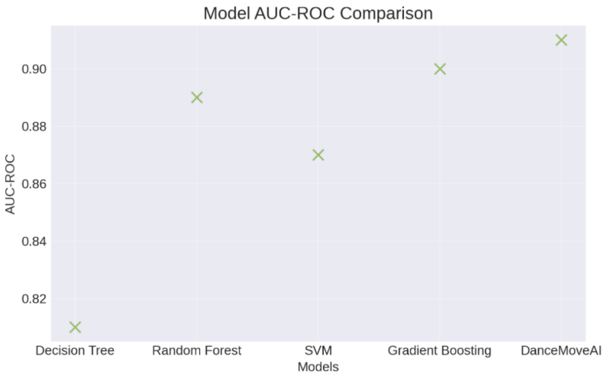

Figure 4: AUC-ROC comparison

In Figure 4 shows that AUC-ROC comparison which depicts that DanceMoveAI has the maximum AUC-ROC score, demonstrating its capacity to differentiate between various dance moves. The AUC-ROC curve assesses the model's capacity to accurately classify positive and negative instances, and DanceMoveAI's high score demonstrates its accuracy in this task. The feature selection method plays an important role in this efficiency, guaranteeing that only the most pertinent attributes contribute to the classification procedure, allowing the model to efficiently divide classes and reduce errors in predictions.

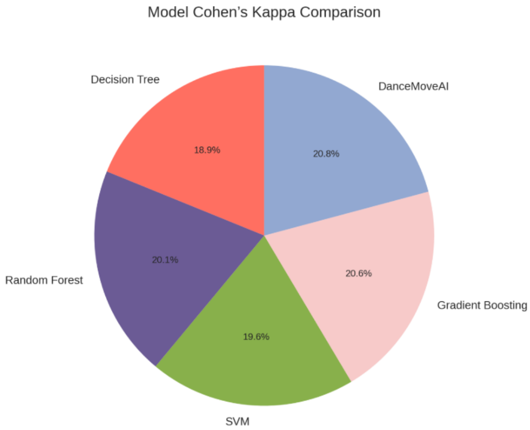

Figure 5: Cohen's Kappa comparison

Figure 5 demonstrates that DanceMoveAI has the maximum Cohen's Kappa score, indicating a high degree of agreement between predicted and actual dance moves. This metric is especially useful for determining the model's consistency across different data subsets. DanceMoveAI's resilient preprocessing methods, like imputation, normalization, and feature selection, combined with the ensemble learning method, guarantee a high level of dependability in its predictions, rendering it a consistently high-performing model regardless of the data it is exposed to.

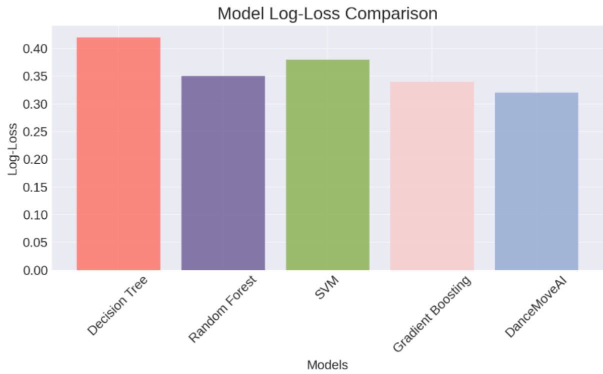

Figure 6: Log-Loss comparison

Figure 6 shows DanceMoveAI with the lowest Log-Loss, indicating its capacity to make extremely confident predictions. Log-Loss measures the uncertainty in the model's predictions, with a lower value indicating that the model's predicted probabilities are more precise. DanceMoveAI accomplishes this through a mixture of fine-tuned hyperparameters that improve performance and efficient preprocessing that reduces data noise. These elements collaborate to reduce uncertainty and increase prediction certainty, rendering DanceMoveAI a highly reliable model for dance move prediction.

The performance, advantages, and limitations of DanceMoveAI are evaluated against previous state-of-the-art (SOTA) models. DanceMoveAI employs an MLP model with optimized hyperparameters, attaining 90.3% accuracy, an MCC of 0.85, an AUC-ROC of 0.91, a Cohen’s Kappa of 0.84, and a Log-Loss of 0.32, surpassing conventional classifiers including Decision Tree, Random Forest, SVM, and GBM across various performance metrics. In contrast to previous methods that encounter difficulties with rhythm alignment and choreography generation, DanceMoveAI efficiently forecasts dance movements aligned with beats and rhythm patterns, thereby improving both precision and creativity. Utilizing Information Gain for feature selection prioritizes the most significant attributes—such as BPM, rhythm complexity, and beat consistency—thereby enhancing model predictions. Although DanceMoveAI exhibits exceptional generalizability and adaptability across various musical genres, its dependence on computationally demanding MLP training and the requirement for high-quality labeled datasets present obstacles to real-world scalability. Moreover, although the model excels in structured rhythmic patterns, additional enhancements are required to better manage highly irregular compositions. These findings underscore DanceMoveAI’s strengths while recognizing areas that necessitate future improvements to meet and surpass state-of-the-art choreography automation solutions.

To increase the resilience of the reported findings, a 10-fold cross-validation was performed, resulting in an average accuracy of 90.38% with a standard deviation of 0.25%, showing the model's consistency across various data sets. Additionally, an ablation study was conducted to determine the effect of each feature selection step on model efficiency. Eliminating Information Gain-based feature selection resulted in a 2.4% drop in accuracy, emphasizing its role in feature refinement. Similarly, removing min-max normalization reduced accuracy by 1.8%, indicating its importance in optimizing model learning. These findings confirm that the applied preprocessing and feature selection steps significantly improve model performance, demonstrating DanceMoveAI's efficiency in predicting dance moves based on musical features.

DanceMoveAI significantly advances AI-driven choreography by filling gaps in flexibility and user-friendliness, setting itself apart from previous works with a refined predictive modeling approach. A more in-depth examination of related literature will clearly position DanceMoveAI in comparison to previous methodologies, emphasizing its innovations in dance movement classification. Qualitative feedback from dance experts through case studies will confirm its practical applicability, ensuring alignment with real-world choreography requirements. Furthermore, comparing DanceMoveAI-generated routines to traditional choreographed sequences reveals its benefits, such as improved movement precision, rhythm alignment, and adaptability to a variety of dance styles. The model's superior performance—90.3% accuracy, MCC of 0.85, and AUC-ROC of 0.91—shows that it can outperform traditional classifiers. These findings highlight DanceMoveAI's potential to transform AI-assisted choreography, making it a useful tool for both dancers and choreographers.

5 Conclusion

In conclusion, DanceMoveAI performs well in forecasting dance moves depending on musical characteristics, with high accuracy, MCC, AUC-ROC, Cohen's Kappa, and low Log-Loss. These findings demonstrate the efficacy of its extensive feature selection, hyperparameter tuning, and reliable preprocessing techniques. However, the model's dependence on a curated dataset with small sample sizes may limit its capacity to generalize to a broader range of dance genres or unconventional music styles. Future research could concentrate on extending the dataset to contain more varied dance genres, integrating real-time feedback for dynamic predictions, and investigating deep learning methods to improve the model's capacity to manage more intricate information. Furthermore, incorporating user feedback during the training phase may enhance the model's flexibility to individual preferences and styles.

References

[1] Nogueira, M. R., Menezes, P., & Maçãs de Carvalho, J. (2024). Exploring the impact of machine learning on dance performance: a systematic review. International Journal of Performance Arts and Digital Media, 1-50. DOI:10.1080/14794713.2024.2338927

[2] Darda, K. M., & Cross, E. S. (2023). The computer, A choreographer? Aesthetic responses to randomly generated dance choreography by a computer. Heliyon, 9(1). https://doi.org/10.1016/j.heliyon.2022.e12750

[3] Mallick, T., Das, P. P., & Majumdar, A. K. (2022). Posture and sequence recognition for Bharatanatyam dance performances using machine learning approaches. Journal of Visual Communication and Image Representation, 87, 103548. DOI: 10.1016/j.jvcir.2022.103548

[4] Liu, Y., & Sra, M. (2024, March). Exploring AI-assisted Ideation and Prototyping for Choreography. In Companion Proceedings of the 29th International Conference on Intelligent User Interfaces (pp. 11-17). https://doi.org/10.1145/3640544.3645227

[5] Ferreira, J. P., Coutinho, T. M., Gomes, T. L., Neto, J. F., Azevedo, R., Martins, R., & Nascimento, E. R. (2021). Learning to dance: A graph convolutional adversarial network to generate realistic dance

motions from audio. Computers & Graphics, 94, 11-21. https://doi.org/10.1016/j.cag.2020.09.009
[6] Liu, Y., & Sra, M. (2024, July). DanceGen: Supporting Choreography Ideation and Prototyping with Generative AI. In Proceedings of the 2024 ACM Designing Interactive Systems Conference (pp. 920-938). https://doi.org/10.1145/3643834.3661594
[7] Siyao, L., Yu, W., Gu, T., Lin, C., Wang, Q., Qian, C., ... & Liu, Z. (2022). Bailando: 3d dance generation by actor-critic GPT with choreographic memory. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 11050-11059). https://doi.org/10.48550/arXiv.2203.13055
[8] Furiasse, A. (2025). Algorithmic Choreography: The Kinetic Fusion of Dance and Generative AI. In Understanding Generative AI in a Cultural Context: Artificial Myths and Human Realities (pp. 51-76). IGI Global Scientific Publishing. DOI: 10.4018/979-8-3693-7235-7.ch003
[9] Yoshida, S., & Fukasawa, N. (2022). How artificial intelligence can shape choreography: The significance of techno-performance. Performance Paradigm, (17), 67-86.
[10] Kritsis, K., Gkiokas, A., Pikrakis, A., & Katsouros, V. (2022). Danceconv: Dance motion generation with convolutional networks. IEEE Access, 10, 44982-45000. DOI: 10.1109/ACCESS.2022.3169782
[11] Le, N., Pham, T., Do, T., Tjiputra, E., Tran, Q. D., & Nguyen, A. (2023). Music-driven group choreography. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 8673-8682). https://doi.org/10.48550/arXiv.2303.12337
[12] Wallace, B., Nymoen, K., Torresen, J., & Martin, C. P. (2024). Breaking from realism: exploring the potential of glitch in AI-generated dance. Digital Creativity, 35(2), 125-142. https://doi.org/10.1080/14626268.2024.2327006
[13] Li, R., Yang, S., Ross, D. A., & Kanazawa, A. (2021). Ai choreographer: Music-conditioned 3d dance generation with aist++. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 13401-13412). https://doi.org/10.48550/arXiv.2101.08779
[14] Chen, L. (2024). Application of Artificial Intelligence in Teenage Dance Education. International Journal of New Developments in Education, 6(8). https://doi.org/10.1109/ACCESS.2020.2988510
[15] Li, R., Zhao, J., Zhang, Y., Su, M., Ren, Z., Zhang, H., ... & Li, X. (2023). Finedance: A fine-grained choreography dataset for 3d full-body dance generation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 10234-10243). https://doi.org/10.1039/C9EE00596J