A comparative Study between three Multi-label Classifiers Machine learning Models for Music Genre Classification

Devesh Prakash Vaid, Dinesh Sheoran, Parveen Kumar

Maharaja Surajmal Institute of Technology, New Delhi, India

Abstract: In this study, we looked at the categorization of musical genres using machine learning. As there are many different genres of music, machine learning multiclass classifier models are used. Each class label is classified by the multiclass in a highly inconsistent manner. To comprehend the problem, we applied popular classifier models, such as the K-Nearest Neighbour Algorithm (KNN), Support Vector Machines (SVM), and Convolutional Neural Network Algorithm, that are utilised in identifying music genres both supervised and unsupervised (CNN). We assessed each classifier model's performance separately and compared it to a function approach we created to enhance multiclass and overall performance. This was accomplished utilising the information we gathered when our classifier models were being trained. With the use of this data, we assigned a score to each classifier for predicting each label genre, and we then utilised these new score metrics to choose the prediction of the classifier that achieved the greatest score. Our results show that by employing this strategy, we were able to get outcomes that were consistent as well as superior. As was previously noted, multiclass classifier models may identify one class more accurately and precisely than the other but can deliver bad performance in other classes, therefore this is extremely beneficial in the actual world. By comparing each classifier model, we were able to improve multiclass performance as well as overall performance.

Index Terms: machine learning, music genre classification, multiclass classification.

1. Introduction

A music genre is a classification system to differentiate different music flavours into different styles. It is the art of combining instrumental and vocal tones in a structured manner that gives the music its distinctive characteristics. As a result, all artistic compositions, that belong to the same classification share some similarities in form or style. This genre classification is used in other forms of arts, including literature, television, cinema, and other forms. It sort pieces of work that fit under a specific type of taste after analyzing the standard features. Digital Music Services provides access to over 40 million songs for streaming and employs a recommendation system that relies on a machine learning classifier to assist users in finding new music. By analyzing music characteristics, the company applies machine learning algorithms to understand users' preferences and suggest comparable music and songs. The commonly adopted approach to categorize musical genres involves assessing song attributes such as rhythm patterns, chord progressions, and instruments employed.

A tremendous world exists in music. There are various genres, subgenres, and styles, ranging from jazz to blues, classical to rap. There are many different genres in our world, but not every genre is as well-liked and prosperous as others. As a result, certain genres are undoubtedly more popular than others in the music industry or market.

In this paper, the following classification models will be used for the analysis:

i. **K-Nearest Neighbour Algorithm (KNN)**
A supervised machine learning approach called the K-Nearest Neighbor method is used to solve classification and regression issues. use previously processed, labelled data to process future,

unlabeled data. The KNN classifier is used to categorise the genres of music; it analyses comparable songs and thinks that because they appear to be close to one another, they belong to the same group. The best results have been obtained using this strategy among the several different classifiers that are prevalent in this idea [1].

ii. **Support Vector Machines (SVM)**

A supervised learning approach called Support Vector Machines (SVM) may be applied to classification and regression analysis issues [2]. SVM is a machine learning model for binary classification that primarily focuses on the separation of data into two class labels at once [3] [4]. We may use the SVM classifier to divide songs into two separate musical genres at once because this approach only allows for the categorization of data into two different class labels. For instance, 10 songs may be analysed at once, and computers with SVM capabilities could sort them into the Hip-Hop or Rock categories. Additionally, Multiclass SVM may be applied in this scenario to simultaneously filter songs into many musical genres.

iii. **Convolutional Neural Network Algorithm (CNN)**

With the Convolutional Neural Network Algorithm (CNN) [5], machines may learn unsupervisedly and predict future data by processing and interpreting visual inputs. Machines maintain the value of these datasets in the form of weights, which subsequently aid in the classification of new information, by analysing photos and utilising visual recognition to learn knowledge. CNN enables machines to analyse spectrograms (visual graphs of music frequencies) as CNN works on pictures, which, in turn, helps computers to determine the category of specific music when it comes to music genre categorization using neural networks.

2. LITERATURE REVIEW

Since the beginning of the Internet, the categorization of music into different genres has received much investigation. In 1999, researchers [6] used supervised machine learning methods, such as Gaussian Mixture models and k-nearest neighbor classifiers, to tackle this issue. They presented three feature sets - timbral structure, rhythmic content, and pitch content - for categorizing music. Hidden Markov Models (HMMs), which are commonly employed for voice recognition tasks, have also been investigated for music genre classification by several researchers [7]-[12]. Other researchers [13] have studied support vector machines (SVMs) with various distance metrics to categorize music genres. Psycho-acoustic elements, such as STFT taken on the Bark Scale, play a significant role in identifying music genres, as highlighted by [14]-[18]. Some of the features first explored by [6] include spectral contrast, spectral roll-off, and mel-frequency cepstral coefficients (MFCCs). Auditory and visual information are used in tandem to train Support Vector Machines and AdaBoost classifiers [19]. Many studies employ these methods and characteristics to analyze speech and other audio data as deep neural networks gain prominence [20]-[25]. However, representing audio signals in the time domain for inputting them into neural networks can be challenging due to their high sampling rate. This issue has been addressed by [26] for audio generation tasks. A research, [27] that employs deep neural networks to compare various musical elements is available. Spectrogram graph representations of a signal, which encode both time and frequency information into an image, are a common alternative feature representation. Convolutional neural networks (CNNs) can be trained using spectrograms, which can be treated as images [28][5][29] used the raw MFCC matrix as input for a CNN that predicted the musical genre for input audio files. A CNN was given a constant Q-transform (CQT) spectrogram as input for classification in [30].

Many studies also done in multi-label genre classification for addressing the different performance across different label or class. For instance, [31] conducted a research in which they reviewed various strategies for unbalanced multi-label classification. A study to overcome the logical label semantics' ambiguity[32].

One key factor that can affect the performance of the multi-label classifiers we are comparing is the use of parallel computing techniques. In particular, the K-Nearest Neighbor (KNN) algorithms that are being used relies heavily on distance calculations between data points, which can be computationally intensive for large datasets [33]. To address this challenge, exploration of speculative parallelization techniques that aim to predict which computations are likely to be needed in the future, and pre-compute them in parallel to reduce overall runtime. While it is still an area of active research, incorporating such techniques into our KNN classifier could potentially improve its performance in the context of music genre classification.

Music genre classification using fuzzy-based clustering [34] is a popular technique for analyzing the data set of ML classifiers. By grouping users based on their preferences and other attributes using fuzzy logic, this method can identify patterns and relationships that might be missed by traditional clustering techniques. This allows the system to create more nuanced and accurate models of user preferences, which can be used to improve the overall effectiveness of music recommendation systems. Additionally, by incorporating user feedback and adjusting the clustering parameters, the system can continue to refine its models and deliver more personalized recommendations over time.

A research [35] [36] conducted on the combination of Software-Defined Networking (SDN) and the Internet of Things (IoT) for music genre classification using ML classifiers. The study suggests that centralizing and programming network resources can lead to improved efficiency and scalability of data transmission and management. Furthermore, integrating IoT devices into the network can provide a range of new data sources that can enhance the accuracy of music genre classification. For instance, IoT sensors can capture data on listener behavior, which can be used to create personalized music recommendation systems. The flexible SDN infrastructure can facilitate rapid prototyping and deployment of these systems, allowing for continuous improvement and adaptation to meet evolving user preferences. Ultimately, this can result in more personalized and effective music recommendations and an improved overall user experience.

Utilizing deep learning techniques in robotics can have a significant impact on music genre classification through ML classifiers [37] [1]. Deep learning offers advanced data analysis and pattern recognition methods that can be applied to music genre classification. Convolutional neural networks (CNNs) can be used to detect patterns in the spectral content of different genres and extract pertinent features for classification, while recurrent neural networks (RNNs) can identify rhythmic and melodic structures by processing audio signals as time-series data. Incorporating these advanced techniques can result in more personalized and accurate music recommendations, ultimately leading to a more satisfying and improved user experience in music recommendation systems. Therefore, deep learning in robotics has the potential to greatly enhance the effectiveness and precision of music genre classification using ML classifiers.

3. SCOPE OF THE WORK

3.1. Problem Statement

As we discussed in our literature review, every study in Music genre Classification is done to discover and compare new techniques but our study is purely focused on finding if techniques can classify one genre more accurately than the others or not. If yes, can we use this data to increase prediction accuracy? In our research, we focused on the three most prevalent methods for music genre classification, namely K-Nearest Neighbour (KNN), Convolutional Neural Network Algorithm (CNN), and Support Vector Machines (SVM). We integrated our discoveries to determine whether there were any enhancements in prediction accuracy.

3.2. Goals of the research Paper

In this paper, we will dig deep into the common classifiers used in prediction of music genre, both supervised and unsupervised, and compare each classifier based on their performance in predicting each label. Then we will implement a function to make prediction based on the performance of each classifiers on each label to improve overall prediction performance and compare it with each classifier.

3.3. Objectives of the research Paper

We will start by implementing preparing our dataset by performing data preprocessing like – cleaning, features extraction, splitting the dataset into test and train sets. Then we will implement the three most common techniques of music genre classification, K-Nearest Neighbour (KNN), Convolutional Neural Network Algorithm (CNN) and Support Vector Machines (SVM) with the help of Machine Learning Python libraries like SciKit-Learn and Keras and train and test each classifier with our prepared dataset.

We required some performance metrics to determine the performance of our classifiers for which we will be using measures like- Accuracy, Recall, Precision, F-Score which are common model evaluation measures. Then we compile and store these measures along with the predictions for later analysis. We will feed these measures and prediction to a function which we will be building to select a model among the three, based on the performance of predicted label by each model. Afterward, we will choose the forecasts made by the model that the function returns. By doing so, we will be relying on three models to make predictions, which could, in theory, enhance the precision, recall, and F-measure of our predictions, and ultimately, elevate the accuracy of the classification process.

4. Materials and Methods

4.1. Dataset Preparation

Our research employs the widely recognized GTZAN dataset, which was also used in a prominent publication on genre classification called "Musical genre classification of audio signals" [6]. The dataset consists of a total of 1000 songs, 30 seconds in length each, with 100 songs per genre. Among the ten represented music genres are rock, jazz, metal, classical, disco, country, blues, pop, and reggae.

4.2. Feature Extraction

Feature extraction refers to the process of converting raw data into numerical features that can be manipulated, while preserving the information contained in the original dataset. This technique is superior to directly applying machine learning to the raw data, as it yields better results. The data in our case are music files that are part of our Dataset. Utilizing librosa, the feature extraction was completed. A Python library for analysing music and audio is called librosa. It offers the components required to develop music information retrieval systems.

1. Content-based Features

Audio signals are split down into smaller parts called frames in content-based fingerprinting. Certain signal characteristics are computed for each frame. These data include the number of zero crossings, the length, the energy, the loudness, the pitch, and more. Once completed, these statistics are utilised to construct a signal fingerprint to determine if two sounds are similar or distinct. Content-based or manually derived characteristics can be divided into frequency and temporal domains.

Time Domain Attributes

These attributes were taken from the unprocessed audio signal.

1. **Central moments:** This consists of the signal's amplitude's mean, standard deviation, skewness, and kurtosis.
2. **Zero Crossing Rate (ZCR):** When a signal changes from a positive to a negative sign from a positivesign, it is said to be at a zero-crossing point. The number of zero-crossings present in each frame of the 10-second signal is calculated after it has been split into smaller frames. The frame length has been set to 2048 points and the hop size is 512 points. Note that all of the features covered in this section have utilised these frame settings consistently. The last representative features picked are the ZCR's mean and standard deviation for all frames.
3. **Root Mean Square Energy (RMSE):** Calculations of a signal's energy are as follows:

$$\sum_{n=1}^{N} |x(n)|^2 \quad (3)$$

The calculation of the root mean square value can also be expressed in the following manner:

$$\sqrt{\frac{1}{N}\sum_{n=1}^{N} |x(n)|^2} \quad (4)$$

After calculating RMSE frame per frame, we average and standardise the results over all frames.

4. **Tempo:** The tempo of a musical piece, typically measured in beats per minute (BPM), is referred to as tempo. It makes sense that different genres of music would have various tempos. To obtain a single tempo value from audio pieces, which can have varying tempos over time, we calculate the mean over several frames. The librosa feature initially computes a tempogram before estimating a single tempo value.

4.3. Train Test Split

With the help of the train-test-split method, we may test the performance of a classifier on fresh or untried data. The process is described as follows:

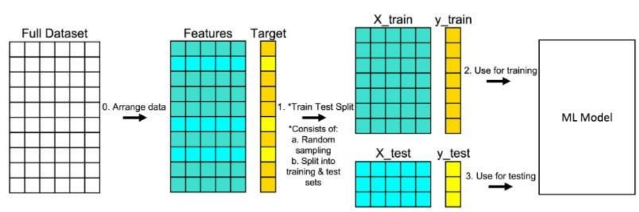

Fig. 1: Train-Test split

1. **Arrange data into features and target**

Data was organised into features and targets using Sckikit-learn. Train test Split by Scikit-Learn divides the data into features and targets. A features matrix in sci-kit-learn is a two-dimensional data grid where the rows correspond to goals and the columns to features. The classes that we wish to predict from the data are a goal. Music Genres serve as our target.

2. **Splitting of dataset into training & testing sets**

To divide the data into training and testing sets, we will utilise the train_test_split module of the SciKit-learn toolkit.

```
from sklearn.model_selection import train_test_split

feature_train, feature_test, target_train, target_test = train_test_split(
    feature_set, target_set, test_size= 0.20, random_state=0)

print("feature_train Set Shape: ", feature_train.shape)
print("feature_test Set Shape: ", feature_test.shape)
print("target_train Set Shape: ", target_train.shape)
print("target_test Set Shape: ", target_test.shape)

feature_train Set Shape:  (800, 57)
feature_test Set Shape:  (200, 57)
target_train Set Shape:  (800,)
target_test Set Shape:  (200,)
```

Fig 2: Python Code for train-test split

For a certain train test split, the columns feature train, feature test, target train, and target test from the original data frame (df) will be used. In the code below, train test split divides the data into training and testing sets, returning a list with four NumPy arrays, while train size = 0.20 divides the data into training and testing sets, each with 80% of the data.

4.4. Implementing Classification Models

Scikit-Learn was used to implement the supervised classifiers, knn and svm. David Cournapeau created the Python module Scikit-Learn for the first time in 2007. It includes a variety of helpful algorithms that are simple to use and modify for classification and other machine learning applications. Before Scikit-Learn can be used, this foundational stack of libraries that SciPy is built upon must be installed.

i. K-Nearest Neighbour Algorithm (KNN)

The knn model is initially imported in the code below. All machine learning models are implemented as Python classes in scikit-learn.

```
from sklearn.neighbors import KNeighborsClassifier

KNN = KNeighborsClassifier(n_neighbors = 5)
KNN.fit(feature_train, target_train)
```

```
KNeighborsClassifier()
```

Fig. 3. Implementation of K-Nearest neighbour Classifier (KNN)

After importing we make an instance of the model and storing it in a variable named KNN which we use to refer our model. After than we are training the model with our training set. It will also store the learned information or weights of the model in the memory. We will use these weights later in testing and later analysis in this paper.

ii. Support Vector Machines (SVM)

similarly in svm, we imported the svm model in the first line of our code.

```
: from sklearn.svm import SVC
  SVM = SVC(kernel='rbf', random_state = 1)
  SVM.fit(feature_train, target_train)
```

```
: SVC(random_state=1)
```

Fig. 4. Implementing Support Vector Machines Classifier (SVM)

Then we make and stored an insctance of the model in a variable named SVM in the next line. Then we trained our model with the training set similary in knn implementation.

iii. Convolutional Neural Network Algorithm (CNN)

Our CNN architecture is comprised of an input layer followed by five convolutional blocks. The CNN architecture we employ involves four convolutional layers and a fully connected layer followed by a softmax classifier [5]. Convolutions are utilized to extract characteristics from the input data, while dropout aids in preventing overfitting. Relu activations, which are non-linear functions used as neuron activation functions, are also used. Following four convolutional layers, the matrix is then flattened, with a dropout probability of 0.5 for regularization. Finally, the last layer consists of a densely connected layer utilizing a 'softmax' activation function to generate class probabilities for each of the ten labels. When given an input, the classifier selects the most probable class from its list of classes.

Categorical cross-entropy (also known as categorical log loss) is shown in equation 1:

$$CE = -\sum_{i}^{C} t_i * \log(s_i) \quad (1)$$

Softmax is the most commonly used activation function in neural networks. The degree to which our predictions deviate from the true values or labels is quantified using cross-entropy loss. The output classes in this scenario are binary (0 or 1), and the accuracy of these classifications can be determined using cross-entropy loss. The model was saved in a variable called CNN, and it was built, compiled, and trained using *Keras* functions.

4.5. Model Testing

Model testing is the process of evaluating the performance of a trained model using a separate testing dataset. To test each we used prediction function of *Scikit-learn* and *keras* for our supervised classifier models, i.e., knn and svm and our unsupervised model, i.e. cnn respectively. Once, we test all the three classifier models, we compile all the predictions in a table matrix. This will be useful for our function we build later for choosing the best classifier for each target label.

4.6. Model Evaluation

We will next move on to model evaluation once each model has been tested. The method through which we measure the calibre of a system's predictions is called model evaluation. To achieve this, we evaluate the performance of the recently trained model on a fresh, unrelated dataset. This model will assess labelled information against its own forecasts.

We calculated the performance metrics for each classifier model using Scikit-built-in learn's methods to arrive at the aforementioned measurement.

iii. Convolutional Neural Network Algorithm (CNN)

Table 1
Performance measures of Convolutional Neural Network Algorithm (CNN)

Genre	precision	recall	f1-score	support
blues	0.8624	0.9106	0.8859	179.00
classical	0.9122	0.9689	0.9397	193.00
country	0.8073	0.8381	0.8224	210.00
disco	0.8596	0.7946	0.8258	185.00
hiphop	0.8350	0.7696	0.8010	217.00
jazz	0.8768	0.9036	0.8900	197.00
metal	0.9103	0.9355	0.9227	217.00
pop	0.8579	0.8895	0.8734	190.00
reggae	0.8462	0.8500	0.8481	220.00
rock	0.8246	0.7421	0.7812	190.00
accuracy	**0.8599**	**0.8599**	**0.8599**	**0.86**
macro avg	**0.8592**	**0.8602**	**0.8590**	**1998.00**
weighted avg	**0.8590**	**0.8599**	**0.8587**	**1998.00**

i. K-Nearest Neighbour Algorithm (KNN)

Table 2
Performance measures of K-Nearest Neighbour Algorithm (KNN)

Genre	precision	recall	f1-score	support
blues	0.8341	0.9832	0.9026	179.00
classical	0.8750	0.9793	0.9242	193.00
country	0.8070	0.8762	0.8402	210.00
disco	0.7655	0.9351	0.8418	185.00
hiphop	0.8744	0.8986	0.8864	217.00
jazz	0.9360	0.8173	0.8726	197.00
metal	0.9760	0.9355	0.9553	217.00
pop	0.9390	0.8105	0.8701	190.00
reggae	0.9567	0.9045	0.9299	220.00
rock	0.9789	0.7316	0.8373	190.00
accuracy	**0.8874**	**0.8874**	**0.8874**	**0.89**
macro avg	**0.8943**	**0.8872**	**0.8860**	**1998.00**
weighted avg	**0.8959**	**0.8874**	**0.8872**	**1998.00**

ii. Support Vector Machines (SVM)

Table 3:
Performance measures of Support Vector Machines (SVM)

Genre	precision	recall	f1-score	support
blues	0.8689	0.8883	0.8785	179.00
classical	0.8692	0.9637	0.9140	193.00
country	0.8390	0.8190	0.8289	210.00
disco	0.8103	0.8541	0.8316	185.00
hiphop	0.8969	0.8018	0.8467	217.00
jazz	0.8429	0.8985	0.8698	197.00
metal	0.9167	0.9124	0.9145	217.00
pop	0.8983	0.8368	0.8665	190.00
reggae	0.8848	0.8727	0.8787	220.00
rock	0.7647	0.7526	0.7586	190.00
accuracy	**0.8599**	**0.8599**	**0.8599**	**0.86**
macro avg	**0.8592**	**0.8600**	**0.8588**	**1998.00**
weighted avg	**0.8607**	**0.8599**	**0.8595**	**1998.00**

4.7. Compiling Performance measures

We need to compile performance measures of all three models in a single table for the function we built in the next step. This function take these performance measures along with predictions of all classifier models to determine the best classifier model for that prediction. We will explain the working of this function in the next step.

Table 4
Compiled performance measure of all Classifier Models per label Genre

Genre	f1-measure			Score		
	CNN	KNN	SVM	CNN	KNN	SVM
blues	0.88587	0.902564	0.878453	20	30	10
classical	0.939698	0.924205	0.914005	30	20	10
country	0.82243	0.840183	0.828916	10	30	20
disco	0.825843	0.841849	0.831579	10	30	20
hiphop	0.800959	0.886364	0.846715	10	30	20
jazz	0.89	0.872629	0.869779	30	20	10
metal	0.922727	0.955294	0.91455	20	30	10
pop	0.873385	0.870056	0.866485	30	20	10
reggae	0.848073	0.929907	0.878719	10	30	20
rock	0.781163	0.837349	0.758621	20	30	10

4.8. Implementing a bestModel Function to select best classifiers per genre

The main goal of this paper is to develop a way to predict to improve the prediction performance of each target label genre. To achieve this we developed a function which takes each model's predictions along with their performance metrics to determine the most accurate prediction between the predictions determined by each classifier model respectively. The code above is the function which determines the best-classifier model combined with another function it makes new predictions by selecting the classifier model. Our function first took the prediction done by each model and compared the scores we gave above and chose the three highest scorer classifiers. If the score of two or more models is equal, It will then consider their f1-measure as a score metric to choose the highest-performing classifier among them. As we already compiled the predictions and performance measures of all the classifier models above, we only need to feed this information into the function we build above.

We also build a function to make predictions as per the best-classifier function results. Then we make new predictions using this function and compile them for the next step, which is evaluating these new predictions.

4.9. Evaluating new predictions determined by our function

Evaluating our newly determined predictions is quite similar to that how we tested our original predictions. We used the same SciKit-learn library to evaluate our models.

Table 5
Performance measures of New Predictions by our function.

Genre	precision	recall	f1-score	support
blues	0.8381	0.9832	0.9049	179.00
classical	0.8957	0.9793	0.9356	193.00
country	0.8725	0.8476	0.8599	210.00
disco	0.8366	0.9135	0.8734	185.00
hiphop	0.8784	0.8986	0.8884	217.00
jazz	0.8985	0.8985	0.8985	197.00
metal	0.9761	0.9401	0.9577	217.00
pop	0.9119	0.9263	0.9191	190.00
reggae	0.9476	0.9045	0.9256	220.00
rock	0.9786	0.7211	0.8303	190.00
accuracy	**0.9009**	**0.9009**	**0.9009**	**0.90**
macro avg	**0.9034**	**0.9013**	**0.8993**	**1998.00**
weighted avg	**0.9049**	**0.9009**	**0.9001**	**1998.00**

5. Results and discussions

After deriving and evaluating fresh predictions, we will now measure and compare their efficacy with that of every classifier. For comparison, we will use classifier models' performance measures as a base to compare with our method. Below are the comparison charts of different measures we achieved with our method:

Table 6:
F1-measures of all classifier models comparison with our function

Genre	f1-score			
	CNN	KNN	SVM	Function
blues	0.8859	0.9026	0.8785	**0.9049**
classical	0.9397	0.9242	0.9140	**0.9356**
country	0.8224	0.8402	0.8289	**0.8599**
disco	0.8258	0.8418	0.8316	**0.8734**
hiphop	0.8010	0.8864	0.8467	**0.8884**
jazz	0.8900	0.8726	0.8698	**0.8985**
metal	0.9227	0.9553	0.9145	**0.9577**
pop	0.8734	0.8701	0.8665	**0.9191**
reggae	0.8481	0.9299	0.8787	**0.9256**
rock	0.7812	0.8373	0.7586	**0.8303**

As we can see, our new predictions determined by our function is performing better in terms of all performance measures. New predictions by our function give better F1-scores for each label class than using a single classifier model for predictions. It means that our function can be working properly for determining the best classifier for each label genre. Also, our scoring algorithm is working fine in scoring classifier models on the base of each class label genre. The AOC is also improved significantly by using this function than using only one classifier model. This implies that the function we built successfully determines each label genre better than using a single classifier.

Table 7:
F1-measures of all classifier models comparison with our function

Model	macro avg	weighted avg	accuracy
CNN	0.86	0.86	**0.86**
KNN	0.89	0.89	**0.89**
SVM	0.86	0.86	**0.86**
Function	0.90	0.90	**0.90**

It is expected that the accuracy achieved by this method is better than using a single classifier as the f1-measure and AOC of the method is improved significantly compared to each classifier.

6. Conclusions

This research paper explores the classification of music genres through the use of machine learning. As the music has multiple genres it is done using multiclass classifier models in machine learning. The multiclass inherently is very inconsistent in classifying each class label. To understand the issue we implemented common classifier models used in classifying music genres both supervised and unsupervised, i.e., K-Nearest Neighbour Algorithm (KNN), Support Vector Machines (SVM) and Convolutional Neural Network Algorithm (CNN). We evaluated the performance of each classifier model individually and compared it with a function method, we developed to improve the multiclass and overall performance. We achieved this by using the data we collected during the training of our classifier models. We used this information to give each classifier a score for predicting each label genre and use these new score metrics to select the prediction of the classifier with the highest achieving score. As we can see in our results, using this method we not only achieved better results but also consistency of our results. This is really helpful in the real world because as mentioned earlier, multiclass classifier models can classify one class more than the other more accurately and precisely but can give poor performance in other classes performance. We managed

to achieve better multiclass as well as an overall performance by using our method of comparing with each classifier model. This same method can be applied in other multi-label classifiers problems for better multi-label performance. An example of a problem we think of is in determining multiple diseases in multiple patients. If the said model predicts one disease very accurately let's say 'tuberculosis' but not 'dengue', the model cannot be implemented in the real world as the wrong predictions can be life-threatening in the situations like these. The more the model is consistent in classifying each label. It can be trusted more. This work is done to overcome this issue in multiclass classification by analyzing their performance in each label to achieve better multiclass classification performance.

References

[1] Gupta, S., Chhabra, A., Agrawal, S., Singh, S.K. (2023). A Comprehensive Comparative Study of Machine Learning Classifiers for Spam Filtering. In: Nedjah, N., Martínez Pérez, G., Gupta, B.B. (eds) International Conference on Cyber Security, Privacy and Networking (ICSPN 2022). ICSPN 2021. Lecture Notes in Networks and Systems, vol 599. Springer, Cham. https://doi.org/10.1007/978-3-031-22018-0_24

[2] Khade, G., Kumar, S., & Bhattacharya, S. (2012, December). Classification of web pages on attractiveness: A supervised learning approach. In 2012 4th International Conference on Intelligent Human Computer Interaction (IHCI) (pp. 1-5). IEEE.

[3] Singh, I., Singh, S. K., Singh, R., & Kumar, S. (2022, May). Efficient loop unrolling factor prediction algorithm using machine learning models. In 2022 3rd International Conference for Emerging Technology (INCET) (pp. 1-8). IEEE.

[4] Mengi, G., Singh, S. K., Kumar, S., Mahto, D., & Sharma, A. (2023, February). Automated Machine Learning (AutoML): The Future of Computational Intelligence. In International Conference on Cyber Security, Privacy and Networking (ICSPN 2022) (pp. 309-317). Cham: Springer International Publishing.

[5] Kaur, P., Singh, S. K., Singh, I., & Kumar, S. (2021, December). Exploring Convolutional Neural Network in Computer Vision-based Image Classification. In International Conference on Smart Systems and Advanced Computing (Syscom-2021).

[6] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals. IEEE Transactions on speech and audio processing 10(5)," p. 293-302, 2002.

[7] N. Scaringell and G. Zoia, "On the modeling of time information for automatic genre recognition systems in audio signals. In ISMIR," p. 666-671, 2005.

[8] Gupta, B. B., Yamaguchi, S., & Agrawal, D. P. (2018). Advances in security and privacy of multimedia big data in mobile and cloud computing. Multimedia Tools and Applications, 77, 9203-9208.

[9] Gopal Mengi; Sudhakar Kumar (2022) Artificial Intelligence and Machine Learning in Healthcare, Insights2Tecinfo, pp. 1, https://insights2techinfo.com/artificial-intelligence-machine-learning-in-healthcare/

[10] Gaurav, A., et al. (2022). A comprehensive survey on machine learning approaches for malware detection in IoT-based enterprise information system. Enterprise Information Systems, 1-25.

[11] H. Soltau, T. Schultz, M. Westphal and A. Waibel, "Recognition of music types. In Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on. IEEE," vol. 2, p. 1137-1140, 1998.

[12] Zhang, Y. et al. (2023). Optimal Revenue Analysis of the Stubborn Mining Based on Markov Decision Process. In: Xu, Y., Yan, H., Teng, H., Cai, J., Li, J. (eds) Machine Learning for Cyber Security. ML4CS 2022. Lecture Notes in Computer Science, vol 13656. Springer, Cham. https://doi.org/10.1007/978-3-031-20099-1_25

[13] M. I. Mandel and D. Ellis, "Song-level features and support vector machines for music classification. In ISMIR," vol. 2005, p. 594-599, 2005.

[14] T. Lidy and A. Rauber, "Evaluation of feature extractors and psycho-acoustic transformations for music genre

classification. In ISMIR," p. 34-41, 2005.
[15] Zhang, Z., et al. (2017). CyVOD: a novel trinity multimedia social network scheme. Multimedia Tools and Applications, 76, 18513-18529.
[16] Muskaan Chopra, Sunil K. Singh (2021), MLOps: A New Era of DevOps, Powered by Machine Learning, Insights2Techinfo, pp. 1, https://insights2techinfo.com/mlops-a-new-era-of-devops-powered-by-machine-learning/
[17] Zou, L., et al. (2019). A novel coverless information hiding method based on the average pixel value of the sub-images. Multimedia tools and applications, 78, 7965-7980.
[18] E. Zwicker and H. Fastl, "Psychoacoustics facts and models," 1999.
[19] L. Nanni, Y. M. Costa, A. Lumini, M. Y. Kim and S. R. Baek, "Combining visual and acoustic features for music genre classification. Expert Systems with Applications 45," p. 108-117, 2016.
[20] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn and D. Yu, "Convolutional neural networks for speech recognition. IEEE/ACM Transactions on audio, speech, and language processing 22(10)," p. 1533-1545, 2014.
[21] Sahoo, S. R., & Gupta, B. B. (2019). Hybrid approach for detection of malicious profiles in twitter. Computers & Electrical Engineering, 76, 65-81.
[22] D. Peraković D, Yadav K, Hsu C (2022) Deep Learning in Healthcare, Insights2Tecinfo, pp.1, https://insights2techinfo.com/deep-learning-in-healthcare/
[23] Cvitić, I, et al. (2021). Ensemble machine learning approach for classification of IoT devices in smart home. International Journal of Machine Learning and Cybernetics, 12(11), 3179-3202.
[24] Bhatti, M. H., et al., (2019). Soft computing-based EEG classification by optimal feature selection and neural networks. IEEE Transactions on Industrial Informatics, 15(10), 5747-5754.
[25] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events. In Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on. IEEE," p. 776-780, 2017.
[26] V. D. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior and K. Kavukcuoglu, "Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499," 2016.
[27] Z. e. a. Qi, "Music genre classification and feature comparison using ml," 2022 7th International Conference on Machine Learning Technologies (ICMLT), 2022.
[28] L. Wyse, "Audio spectrogram representations for processing with convolutional neural networks. arXiv preprint arXiv:1706.09559," 2017.
[29] T. L. Li, A. B. Chan and A. Chun, "Automatic musical pattern feature extraction using convolutional neural network. In Proc. Int. Conf. Data Mining and Applications," 2010.
[30] T. Lidy and A. Schindler, "Parallel convolutional neural networks for music genre and mood classification. MIREX2016," 2016.
[31] A. N. M. G. K. M. Tarekegn, "A review of methods for imbalanced multi-label classification." Pattern Recognition," Pattern Recognition 118, p. 107965, 2021.
[32] T. e. a. Zhao, "Selective label enhancement for multi-label classification based on three-way decisions.," International Journal of Approximate Reasoning 150, pp. 172-187, 2022.
[33] Kumar, S., Singh, S. K., Aggarwal, N., Gupta, B. B., Alhalabi, W., & Band, S. S. (2022). An efficient hardware supported and parallelization architecture for intelligent systems to overcome speculative overheads. International Journal of Intelligent Systems, 37(12), 11764-11790.
[34] A. Sharma et al., "Fuzzy Based Clustering of Consumers' Big Data in Industrial Applications," 2023 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 2023, pp. 01-03, doi: 10.1109/ICCE56470.2023.10043451.
[35] Singh, S. K., Sharma, S. K., Singla, D., & Gill, S. S. (2022). Evolving Requirements and Application of SDN and IoT in the Context of Industry 4.0, Blockchain and Artificial Intelligence. Software Defined Networks: Architecture and Applications, 427-496.
[36] Singh, R., Singh, S. K., Kumar, S., & Gill, S. S. (2022). SDN-Aided Edge Computing-Enabled AI for IoT and Smart Cities. In SDN-Supported Edge-Cloud Interplay for Next Generation Internet of Things (pp. 41-70). Chapman and Hall/CRC.
[37] Aggarwal, K., Singh, S. K., Chopra, M., Kumar, S., & Colace, F. (2022). Deep learning in robotics for strengthening industry 4.0.: opportunities, challenges and future directions. Robotics and AI for Cybersecurity and Critical Infrastructure in Smart Cities, 1-19.