ISSN 2278-3091

Volume 10, No.3, May - June 2021

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse641032021.pdf

https://doi.org/10.30534/ijatcse/2021/641032021

Music Genre Classification using Optimized Sequential Neural Network

Anahitaa R[1], Roopikha Ganesh[2], R Shivani[3], Pattabiraman V[4]

[1]Vellore Institute of Technology, Chennai, India, anahitaa.radhakrishnan2018@vitstudent.ac.in

[2]Vellore Institute of Technology, Chennai, India, roopikha.ganesh2018@vitstudent.ac.in

[3]Vellore Institute of Technology, Chennai, India, r.shivani2018@vitstudent.ac.in

[4]Vellore Institute of Technology, Chennai, India, pattabiraman.v@vit.ac.in

ABSTRACT

Music makes up a huge portion of the contents stored and used over the internet, with several sites and applications developed solely to provide music-related services to their users/ customers. Some of the most challenging tasks in this scenario would include music classification based on languages and genres, playlist suggestions based on music history, song suggestions based on playlist contents, top genres / songs based on listeners' rating, likes, number of streams, song loops, popularity of artists based on number of songs released per year, hit songs per year, etc. One of the most important stages to solve the above-mentioned challenges would be music genre classification. It would be impractical to analyze each and every song in a given database to identify and classify music genres, even though human beings are better at performing such tasks. Hence, useful Machine Learning algorithms and Deep Learning approaches may be used for accomplishing such tasks with ease. A thorough analysis to understand the different uses of Machine Learning and Deep Learning algorithms and relevance of such algorithms with respect to situations would be made to highlight and contrast the advantages and disadvantages of each approach. The outcomes of the optimized models would be visualized and compared to the expected outcomes for better perception.

Key words: Music Genre, Machine Learning, Deep learning, Sequential Neural Networks

1. INTRODUCTION

Classification of music based on genre is easy to do manually and can achieve a considerable level of accuracy. However, it becomes difficult to do the same on a large scale, especially across service platforms where the demand for music is high. Although there are several popular existing machine learning algorithms to solve this issue, they have particularly low accuracy in classifying music based on genres, especially when there are innumerable genres to consider with the rise of different categories and subcategories too. Several Machine learning approaches may also fail without appropriate training, due to the fact that music can be possibly classified into several genres and at times, the incorrect genre. Our objective is to evaluate different approaches to classify music according to their genres and optimize their performance. Given the music files and their respective image files, they are to be grouped into different genres. Classification is done using different approaches for better understanding. The merits, demerits and accuracies are compared and contrasted. Finally, we arrive at a conclusion regarding the solutions considered for this problem statement. This can be implemented in music applications in devices and by music related services to improve the quality of their service.

Some of the constraints pertaining to the approaches considered are also kept in mind. For instance, some songs may belong to several genres (eg: pop and rock). So, in order to eliminate the ambiguity, approaches like KNN may be used to group them into the most relevant genre. Other algorithms which produce different results may be considered for optimization to remove such ambiguities. Other hardware constraints are also considered for this analysis. Classification models with enhanced accuracy and efficiency may be evaluated and their respective outcomes, tabulated and visualized. This provides a better overview regarding the best approaches to obtain maximum accuracy in classification results, with hardware and software constraints taken into consideration. Optimized algorithms prove to be highly efficient as they provide better results whilst consuming a minimum amount of time for effective computations.

The dataset used for our analysis is GTZAN Dataset. It consists of 30 second audio tracks, image files of the audio tracks and 2 csv files. The individual audio tracks are 30 seconds long. Each csv file contains data based on features of songs such as genre labels, track name, chroma feature values, etc. About 60 columns and 9990 observations can be found in one csv file, while the other contains the same number of features, but 1000 observations recorded. Each song is labelled as one among 10 different genres. The image files serve as visual representations of each audio track, mostly plotted based on mel spectrogram.

2. RELATED WORKS

In a work by Beici Liang and Minwei Gu [2], a similar problem of audio-based classification of 11 western music genres, including Rock, Pop, Rap, Country, Folk, Metal, Jazz, Blues, R&B, Electronic Music and Classical Music was taken up and was done using a transfer learning approach. An advantage of this approach is that the pre-trained models that were obtained from the source task of music auto-tagging can be successfully adapted to the target task in order to achieve high performance measurements. However, one disadvantage of this approach was that it was seen that some music genres like R&B were easily misclassified into Pop, Rap, Electronic Music, etc because it encompasses elements from these genres when the transfer learning approach was used.

Keunwoo Choi et al., 2017 [3] used a similar approach for music classification and regression tasks using a pre-trained convnet feature on six music information retrieval and audio-related tasks. Unlike the common approach in transfer learning, they proposed to use the features from every convolutional layer after applying an average-pooling to reduce their feature map sizes. There was a good performance due to the pre-trained convnet feature. baseline MFCC feature for all the six tasks for all the six tasks were outperformed. However, a disadvantage observed is that for five of the tasks, there was no improvement on catenating MFCC features onto convnet features.

In a work by B. Liang et al. [4], transfer learning has been considered a better yet complex procedure to train various new models than machine learning algorithms. It's a knowledge-based approach that trains new models which may not have a proper training dataset based on the existing models by deriving logic between various attributes. Earlier classifications of music genre were done by K. Simonyan and A. Zisserman [13] using CNN based on research by K. Choi et al. [12] auto tagging methods of the audio files to train the model. B. Liang et al. [4] had used a transfer learning approach which was based on the knowledge acquired from these pre-trained and auto-tagged CNN models which were further applied to the target task for music genre classification. The accuracy obtained was around 89% for the above approach. The transfer learning approach used in [4] enables the use for various classification and regression models, for instance we see that in the paper the features that were extracted and obtained for testing from the pre-trained CNNs, were actually trained using a large dataset in the source task. Thus, the obtained features were used in the target task for classification of the recordings which had proved to be a great advantage. But it can be observed in [4] that the use of transfer learning required them to create a very complex algorithm and the search for right models for creating a training dataset that shall be using knowledge-based training makes it way more complex. Deciding on the right features to be extracted to the logical derivations, one may find it too complex and complicated to understand and using it for future work might be quite an effort since understanding it can be quite difficult before implementation.

J. A. Suykens and J. Vandewalle [5] dealt with the least squares version of support for vector machine classifiers. It has been viewed that this approach requires the formation of a set of linear equations in-order to meet the equality contained for a given quadratic programming model. Mercer's condition is also applied for SVM's. Further they have illustrated the least squares SVM model with RBF kernel for compensating the complexity with an efficient generalization, performance with a low computational cost for a two spiral classification problem. The paper [5] formulates a least squares version of the classic SVM's classification problems with 2 classes. For the first on which is a [5] function estimation problem, a support vector interpretation of ridge regression, it considers only equality type constraints instead of the inequalities which is actually found in the classical SVM approach. This is a great advantage since considering the equality constraints for the classification problem enables the formulation in least squares sense. Hence, as a result, [5] instead of quadratic programming the linear equations are directly solved. Though classical SVM's support many values that are zero but, in this method the support values are proportional to the errors. Further in [5] they have reviewed some basic work with regard to support vectors machine classifiers. The least square methods they have discussed deals with only linear equations and fails to implement the inequalities measures.

Detailed comparative studies on the performance of various timbre modelling for unexplored tasks like guitar-playing techniques classification have been performed in the paper Sparse cepstral, phase codes for guitar playing technique classification, by L. Su, L.-F. Yu, and Y.-H. Yang [6], keeping in mind that a practical interpretable automatic transcription can be expected to provide information regarding the playing techniques, apart from the pitch, onset, etc. The study is conducted on several features and phase derivatives (like group delay function or GDF) to efficiently classify electric guitar recordings into different 'playing techniques', essentially very similar to classifying music recordings into different genres. In Electric guitar playing technique detection in real-world recording based on f0 sequence pattern recognition, by Y.-P. Chen, L. Su, and Y.-H. Yang [7] the authors primarily focus on classification of guitar playing techniques like bend and vibrato and also measures the performance of the guitar and real-world electric guitars as well. Though the existing works were based on only single note classification the method proposed in [7] is applicable for classification for single solo tracks as well.

The analysis was done based on a new dataset which contains [7] 42 electric guitar sound solo tracks. Based on the average F-score obtained which was around 74%, it was considered the best under the two-fold cross validation.

In a work by P.-C. Li [8], automatic expressive musical term classification has been taken into consideration to study the interpretational factors. Around 10 main musical terms have been used to compile a new dataset for the instrument, which leads to significantly higher classification accuracy. The authors of the paper came up with an approach which analysed the interpretational factors of expressive musical terms used in the new dataset, which consists of much more expressive interpretations of violin solos, leading to better results compared to the traditional dataset used. Contrast of feature values between expressive and non-expressive performance is found critical in modelling musical expression. The advantages include enhanced efficiency as the dataset contains feature-engineered attributes which ease computations and provide highly accurate classification outputs. At the same time, it is possible to miss out a key feature which could be vital in certain computing scenarios, though its correlation to accurate outputs may be incorrect. Existing errors in the dataset may also contribute to grave errors. But all-in-all, the approach used is effective as long as the dataset used initially is reliable at least to a considerable extent.

An algorithm called Adam is introduced by D. P. Kingma and J. Ba [10]. It is a first-order gradient-based optimization approach for stochastic functions. This method is pretty much easy to implement due to its straight-forward approach, is efficient in performance and uses little memory resources. Due to the above stated merits, this method is also appropriate for scenarios that deal with noisy or sparse gradients. According to the empirical evaluation, Adam works well in practice and provides favorable results. The efficiency of another variant of Adam based on infinity-norm called AdaMax is also discussed here. The algorithms may need further optimizations and relevant evaluation methods, as it would greatly impact the performance. Furthermore, their relevance must be studied in much more detail.

In a research by M Mubasher Hassan et. al. [16], header-based spam filtering was done using machine learning approach. It was observed that an overall improvement in spam filtering is achieved when the naïve-bayes classification technique is applied using an intelligent machine learning based dataset encompassing certain header field-based attributes. An advantage of this method is that, the time-consuming and rigorous filtering done on the body of the email can be made simple without affecting the spam filter performance. Similar classification techniques can be applied in music classification too. However, a disadvantage of this method is that, while it applies for text-based classification, it may not be suitable for audio-based classification.

3. METHODOLOGY

In order to tackle the problem stated previously, gtzan dataset has been used. Several Machine Learning [11] and Deep Learning [9][15] approaches have been considered and optimized for better analysis and implemented in order to evaluate their performance efficiency. Some of the most noteworthy results obtained have been included and highlighted.

KNN (K-Nearest Neighbours) is a lazy algorithm which uses data from a database (separated into different categories or classes) to predict classification of a new sample input.

- After importing the required dataset and performing some basic feature engineering techniques, we choose an optimum value of 'k' to work with (3 in our case).
- Then, the euclidean distance for each of the test data is calculated and sorted in ascending order.
- First 'k' rows are picked from the sorted array and classes are assigned based on frequency of the rows.

The confusion matrix in Figure 1 is one obtained using knn algorithm for the gtzan dataset.

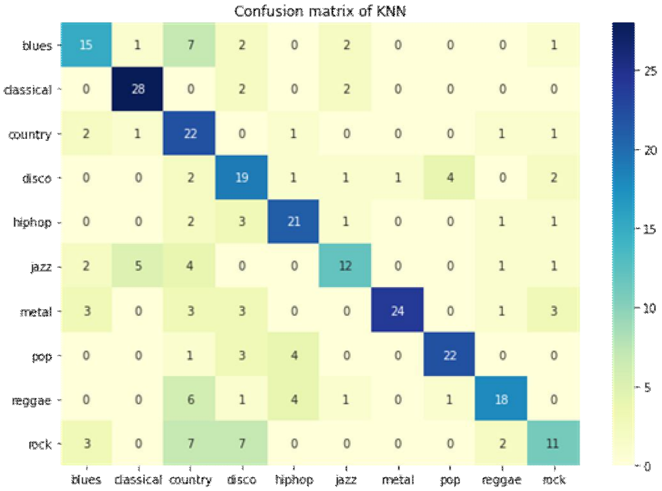


Figure 1: Confusion matrix of KNN

SVC (Support Vector Classifier) is a part of SVM (Support Vector Machine) which returns the best-fit hyperplane based on training inputs which in return, categorizes input data. We may consider linear, polynomial or exponential approaches for this type of classification.

- The dataset (after cleaning) is split into test data and training data.
- Training data is used to train the SVC model, which takes in data as input and produces a hyperplane as an output which best separates the different 'classes' or 'genres'.
- This model is tested for accuracy using the test data.

Figure 2 illustrates the confusion matrix of this algorithm after implementation.

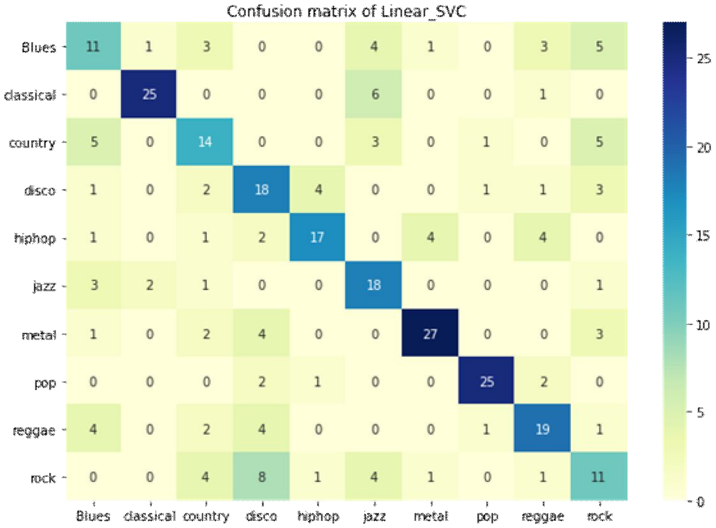


Figure 2: Confusion matrix of Linear SVC

Logistic regression is another classification algorithm which uses sigmoidal functions and probability.

- The training data obtained by splitting the clean dataset is used to fit the logistic regression model.
- Logistic regression returns a probability value obtained using the logistic sigmoid function (1), which is used to segregate data into different discrete classes.

The sigmoid function is represented as :

$$S(z)= 1(1+ e-z) \quad (1)$$

Where S(z): output between 0 and 1, z: input function (algorithm's prediction), e: base of natural log

The plotted confusion matrix as shown in Figure 3 provides a better perspective regarding the efficiency of this approach based on evaluated accuracy of the model.

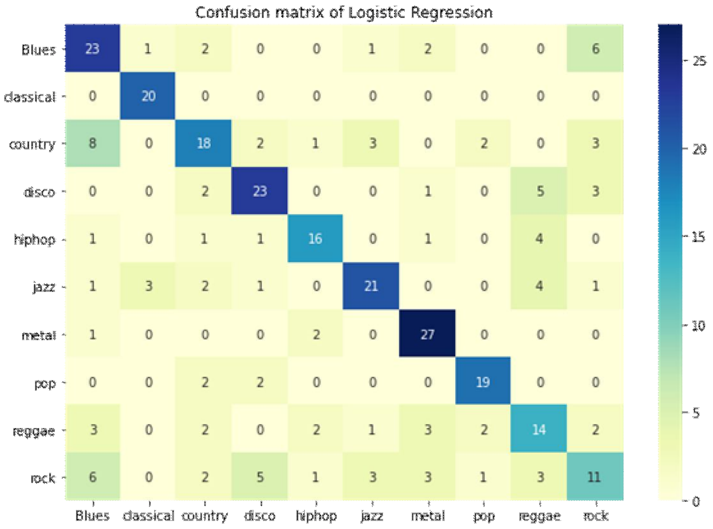


Figure 3: Confusion matrix of Logistic Regression

The scatterplot between chromagram and rms values coloured according to music genres is shown below in Figure 4. Figure 5 depicts the same between mfcc1 and mfcc2.

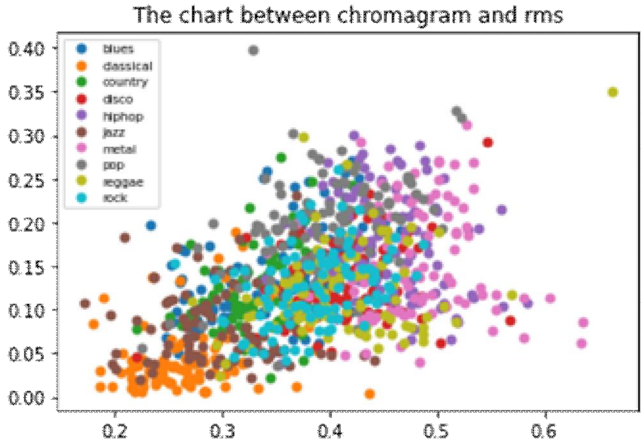


Figure 4: The chart between chromagram and rms

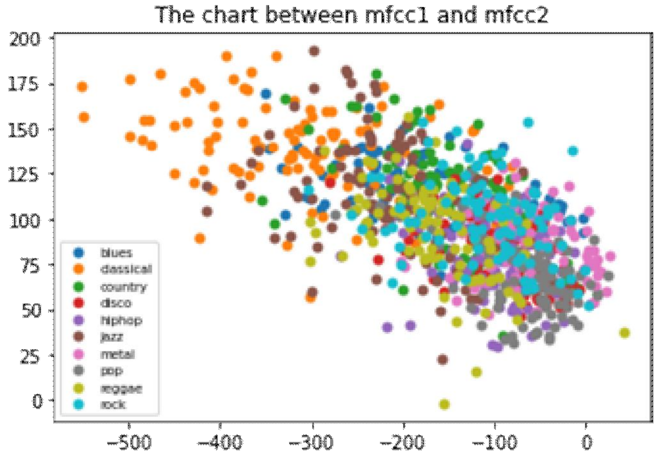


Figure 5: The chart between MFCC1 and MFCC2

To get a stable and more accurate prediction, random forest builds more than one decision tree and merges them together. This algorithm has approximately the same hyperparameters as a decision tree. Pseudocode for Random Forest:

- "k" features are randomly selected from the list of "m" available features (k << m).
- Using the best split point, the node "d" is calculated among the "k" features.
- Using the best split, the node is split into daughter nodes.
- Until "l" nodes are reached, the last three steps are repeated
- To create "n" number of trees, a forest is built by repeating the last four steps "n" number times.

Figure 6 shows the confusion matrix for the Random forest algorithm.

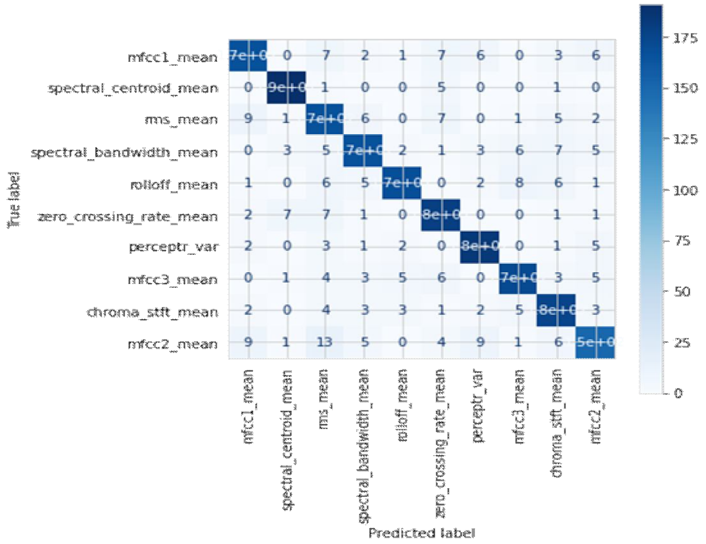


Figure 6: Confusion Matrix for Random forest

XGBoost is a decision-tree-based ensemble Machine Learning algorithm. It uses a gradient boosting framework that helps solve prediction problems involving unstructured data (text, images, etc.). Audio files are unstructured too. Figure 7 shows the confusion matrix for XGBoost algorithm. Hence, XGB is suitable.

XGBoost has the following methods:

- The first method is "Gain", which deals with the total gain that a feature brings on its branches.
- "Frequency"/"Weight" is the second method. The comparative number of times a specific feature occurs in every split of the model's trees is calculated by this method. For features with greater number of categories, this method may be biased.
- XGBoost has one more method, "Coverage". This method takes into consideration the relative number of observations with regard to a feature. For each of the features, the count of observations is recorded and used to select the leaf node.

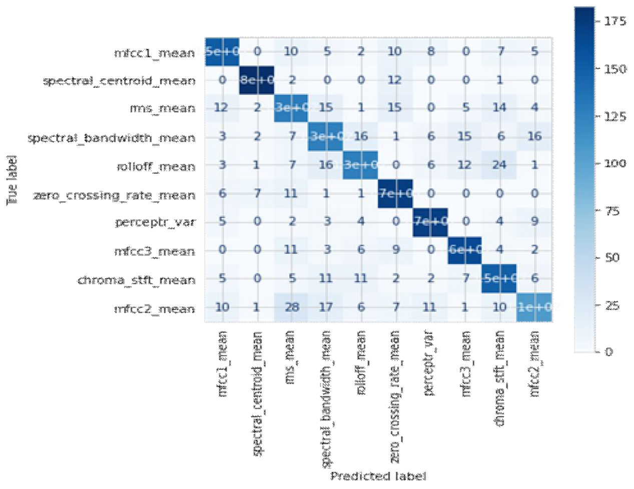


Figure 7: XGB

CatBoost is another algorithm that uses gradient boosting on decision trees. The only boosting algorithm with very less prediction time is CatBoost. This is because of its symmetric tree structure. Figure 8 shows the confusion matrix for CatBoost Algorithm. There are two methods in CatBoost. They are "PredictionValuesChange" and "LossFunctionChange".

In the "PredictionValuesChange" method:

- If the feature value changes, PredictionValuesChange shows the average of prediction changes for each value
- If a greater change in the predicted value is caused by a change in the feature value then the feature has a higher importance.
- For non-ranking metrics, this is the default calculation method for feature importance.

In the "LossFunctionChange" method:

- This is particularly suitable for ranking models. Here, the difference between the loss value of the model with and without the feature is represented by the value for each feature.
- Using the original model, this model is built after this feature is removed from every tree in the ensemble. This is because, without one of the features, retraining the model is computationally costly.

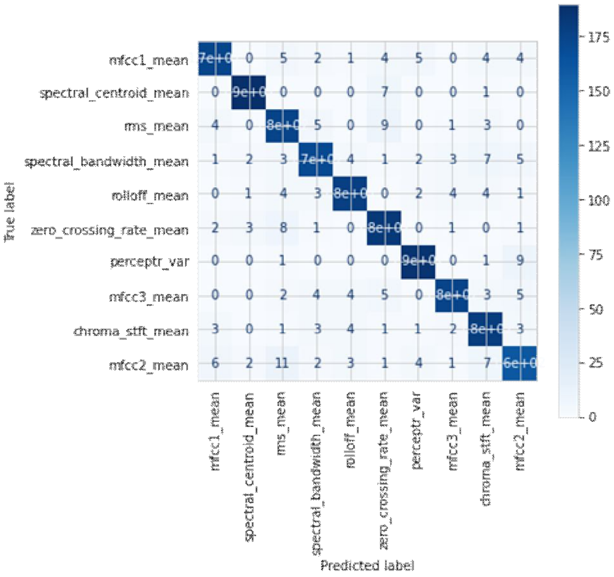


Figure 8: CatBoost

A Sequential model is appropriate for a plain stack of layers where each layer has exactly one input tensor and one output tensor**.** Sequential model by passing a list of layers to the Sequential constructor. Its layers are accessible via the layers attribute. Sequence models are the machine learning models that input or output sequences of data. Sequential data includes text streams, audio clips, video clips, and time-series data.

In our approach we have considered the classical SNN model and have optimised it using 5 different approaches.

- For the first approach we have tried to train the model layer by layer using the keras model. For training the dataset we have divided the dataset into 3 dense layers and have trained them separating based and density and processed with adam optimiser with an epoch value of 70. The epoch value here signifies the number of times the layers are processed ,extracted and trained repetitively to increase the accuracy.
- For our second approach we have segregated our dataset into 4 dense layers and again trained it using keras model of SNN. But this time we raised the epoch value to 100 inorder to to increase the accuracy and efficiency with the same adam optimiser and trained the model.
- For the third approach we have segregated into 5 layers this time and used the keras model for training with SNN approach. But this we used the sgd optimiser with epoch value of 700 for better processing and then trained the model.
- For the fourth approach we segregated into 4 layers and used keras model for training in SNN. But this time we had used the rmspop optimiser with epoch value of 500 and trained the model.
- The fifth approach that we have proposed was based on more coefficient method of segregating the genres. For this method we used the audio files signals, sample rates and amplitudes as the attributes for feature extraction from the dataset. Once the feature extraction was performed using the rmspop optimizer we copied the new modified dataset set into a new excel sheet. Now using this as our training dataset we processed each and every audio files spectrograms value to segregate them and split them up for the test dataset.

Lastly based on the output obtained for the test dataset, we used keras classifier to measure the test dataset accuracy with respect to the training and we achieved the maximum accuracy of all the proposed systems.

4. EVALUATION

In this section we have visualized the results of some optimised SNN models that we have proposed by varying some of the features like epoch value and using different optimisers to see which yields a better accuracy for the classification.

The below Figure 9, Figure 10, Figure 11 and Figure 12 show the outputs for epoch vs accuracy depicting the loss and accuracy lines for different optimisers like [10] adam and sgd and epoch values.

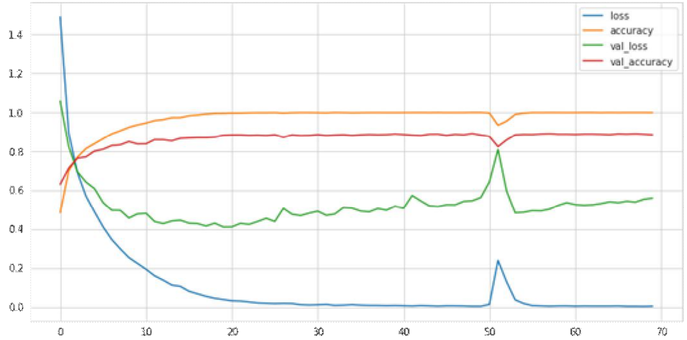


Figure 9: Model 1 SNN (optimizer= adam, epoch=70)

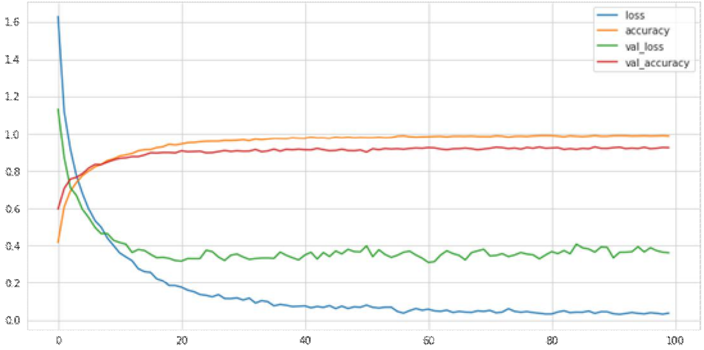


Figure 10: Model 2 SNN (optimizer= adam, epoch= 100)

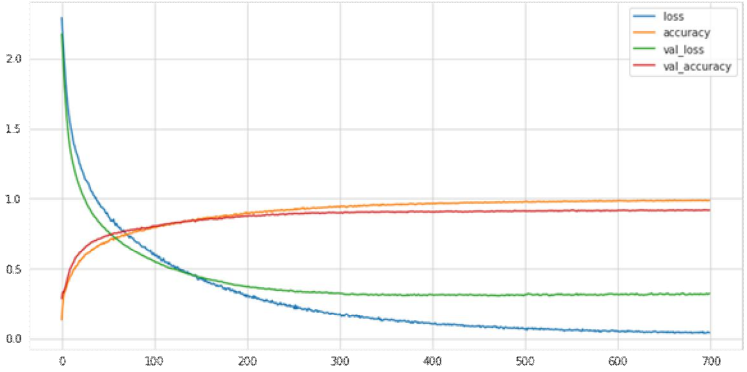


Figure 11: Model 3 SNN (optimizer= sgd, epoch=700)

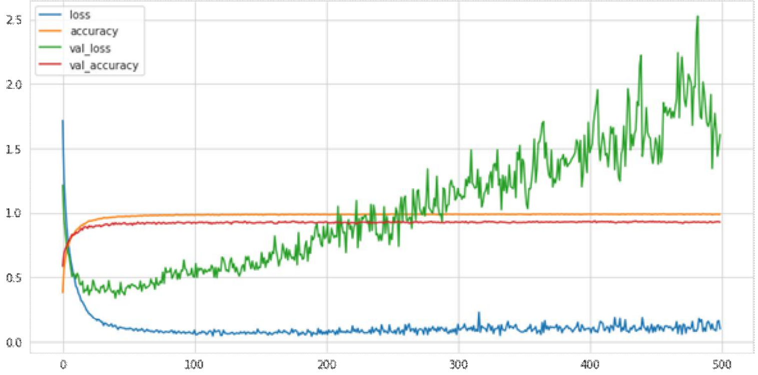


Figure 12: Model 4 SNN (optimizer= rmsprop, epoch= 500)

We have also implemented SNN algorithm using the More Coefficient approach which is another optimised algorithm which segregates and trains the model based on various classification of the coefficient for each feature. Figure 13 visualizes the loss vs accuracy value for this approach.

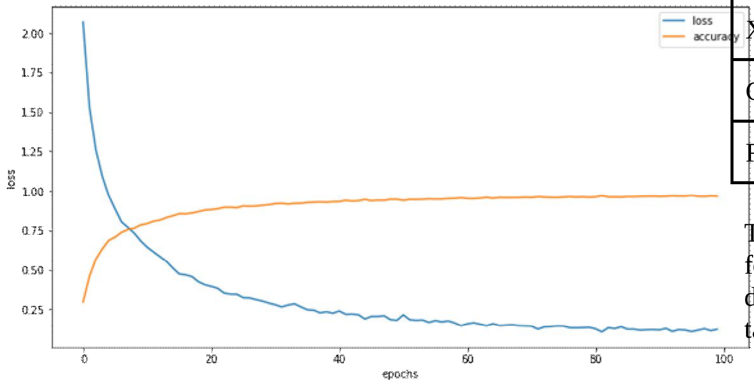


Figure 13: SNN (More Coefficient Approach)

The accuracy of every ML algorithm implemented has been obtained and they have been tabulated in Table 1.

Table 1: Machine Learning Algorithms

Algorithm	**Accuracy**
Knn	60%
Svm (Linear)	62%
Svm (Rbf)	63%
Svm (Polynomial)	62%
Logistic Regression	64%
Logistic Regression (Chroma as a feature)	15.6%
Svm Linear(Chroma as a feature)	16%
Svm Rbf (Choma as a feature)	15%
Polynomial SVM (Choma as a feature)	9%
Knn (Choma as a feature)	19%
Logistic Regression (mfcc as a feature)	54%
Svm Linear (mfcc as a feature)	50%
Svm Rbf (mfcc as a feature)	58%
Polynomial Svm (mfcc as a feature)	54%
Knn (mfcc as a feature)	56%
Xgb	88.02%
CatBoost	89.59%
Random Forest	89.68%

The accuracies of Simple Sequential Neural Network and the four optimized Sequential Neural Network approaches with different optimizers and varying epoch values have been tabulated in Table 2.

Table 2: Sequential Neural Network

Algorithm	**Accuracy**
Simple Sequential Neural Network	88.4%
SNN (Model 1) Optimizer= Adam, Epoch=70	89.13%
SNN (Model 2) Optimizer= Adam, Epoch=100	92.77%
SNN (Model 3) Optimizer= sgd, Epoch=700	92.06%
SNN (Model 4) Optimizer= rmsprop, Epoch=500	93.83%

Table 3 shows the accuracies of Simple Sequential Neural Network, the more coefficient approach and the best accuracy of optimized Sequential Neural Network.

Table 3 : Sequential Neural Network (DIfferent Approaches)

Algorithm	**Accuracy**
Simple Sequential Neural Network	88.4%
Optimized SNN (best accuracy)	92.24%
SNN (more coeff approach)	93.88%

5. RESULTS AND DISCUSSION

The results of implementation of various algorithms have been visualized using bar charts and have been compared.

Figure 14 shows a bar chart comparing the accuracies obtained for the various Machine Learning Algorithms. It can be seen that XGB, Random forest and CatBoost are the algorithms with the highest accuracies.

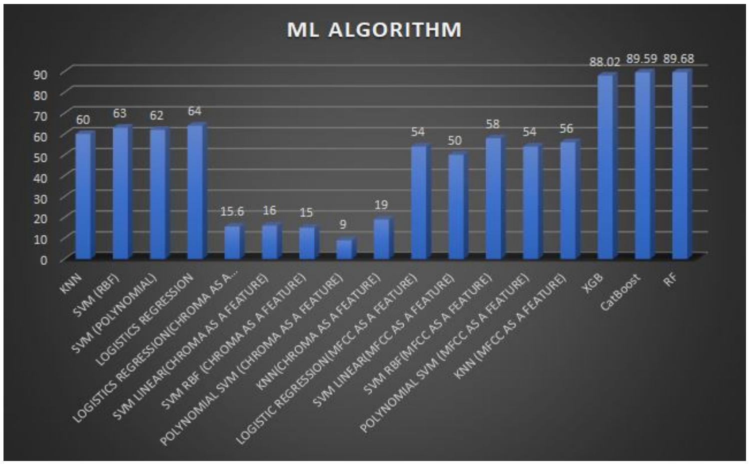


Figure 14: ML Algorithms Accuracies

The accuracies of Simple Sequential Neural Network and the four optimized Sequential Neural Network approaches with different optimizers and varying epoch values have been visualized in Figure 15.

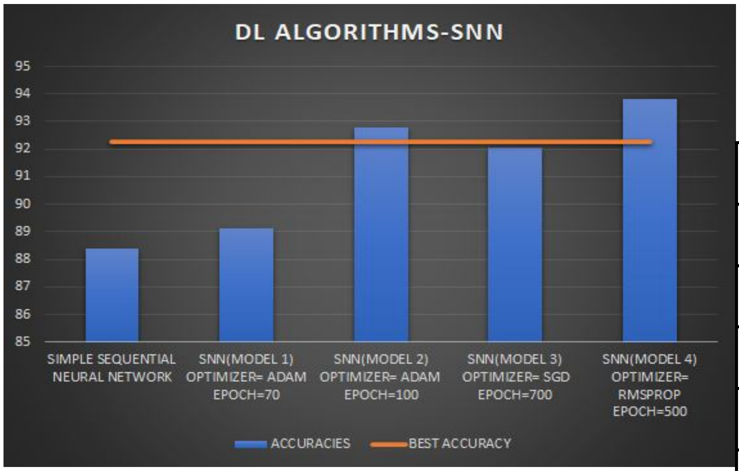


Figure 15: DL Algorithms Accuracies - SNN

Figure 16 shows a bar chart with accuracies of Simple Sequential Neural Network, the more coefficient approach and the best accuracy of optimized Sequential Neural Network.

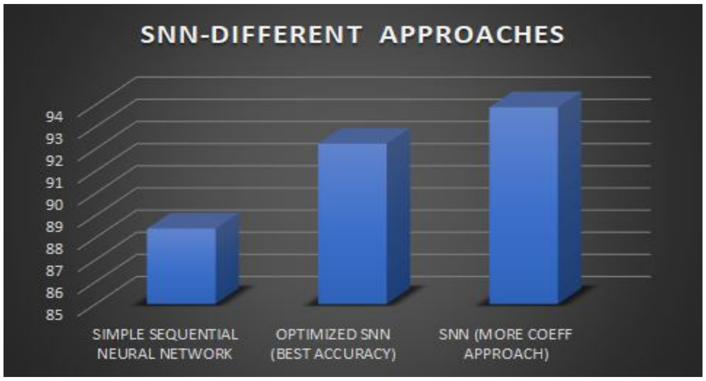


Figure 16: SNN Accuracies - Different approaches

Figure 17 is a visualization of the best ML and DL approaches with their accuracies that we had obtained during our implementation. We find that the optimised SNN algorithms that we proposed show better accuracy than the traditional Machine learning algorithm and the classical SNN approach.

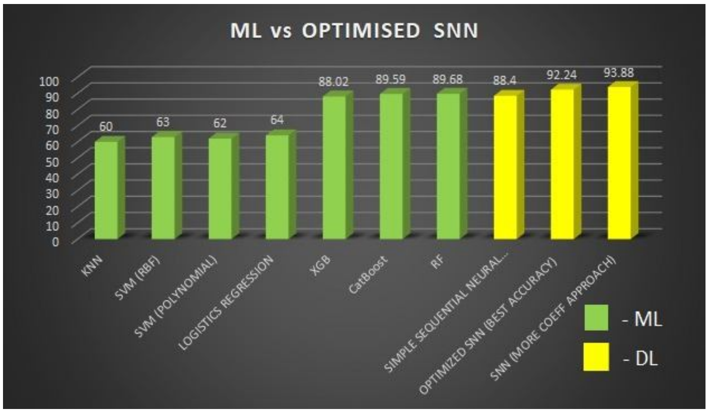


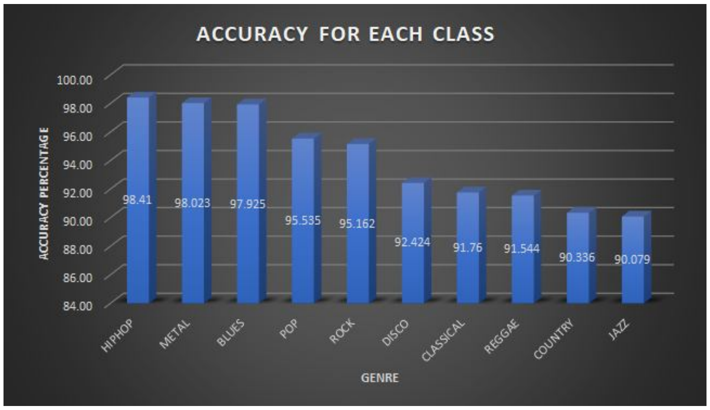
Figure 17: Accuracies of the best ML and DL algorithms

Using the optimized SNN model, the test data have been classified and the accuracies with regard to specific genres have been obtained and tabulated as shown below in table 4.

Table 4: Accuracy for each class

Genre	Accuracy
Hiphop	98.41%
Metal	98.023%
Blues	97.925%
Pop	95.535%
Rock	95.162%
Disco	92.424%
Classical	91.76%
Reggae	91.544%
Country	90.336%
Jazz	90.079%

Figure 18. shows the visual representation of accuracy for each genre based on the optimised sequential neural network using the more coefficient approach



T 18: Accuracy for each class

6.CONCLUSION

After optimization and thorough analysis of several approaches, we obtain the individual best performance accuracies of different models for the gtzan dataset. According to the observations made, Machine Learning approaches are considered to be a bit more time-efficient and require less hardware specifications too, but their accuracies were found to be mediocre. When specific features were considered, they yielded different results too. Uncorrelated features produced undesirable results and perfectly correlated features optimized the algorithms. Feature Engineering approaches had a direct impact on results. Deep Learning Sequential Neural Networks on the other hand turned out to be extremely efficient, yielding high accuracy which depended on several input features, epochs specified and optimizers used. Although they produced desirable results, they were quite time consuming compared to traditional Machine Learning approaches. Hardware limitations have an impact on computation time and thus, directly affect the efficiency. Thus, according to availability of hardware resources and time constraints, the respective models may be considered for implementation. Future works may include approaching this problem using transfer learning techniques [1], which are extremely efficient when modelled correctly. Furthermore, transfer learning is gaining more attention these days and can be used as an effective alternative solution. Several optimization techniques can be applied too. In the deep learning field, RNNs [14] and CNNs [13] can be optimized further to achieve more precision.

ACKNOWLEDGEMENT

We would like to express our deep and sincere gratitude to our Professor, Pattabiraman V, for guiding and mentoring us throughout the journey of our research and making the implementation of this project possible with his knowledge and expertise in the field.

REFERENCES

1. S. J. Pan and Q. Yang, “**A survey on transfer learning,**” *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345– 1359, 2010
2. B. Liang and M. Gu, "**Music Genre Classification Using Transfer Learning**," *2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), 2020*, pp. 392-393, doi: 10.1109/MIPR49039.2020.00085.
3. K. Choi, G. Fazekas, M. Sandler, and K. Cho, “**Transfer learning for music classification and regression tasks**,” in Proceedings of the *18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 141–149
4. B. Liang, G. Fazekas, and M. Sandler, “**Transfer learning for piano sustain-pedal detection**,” in *2019 International Joint Conference on Neural Networks (IJCNN). IEEE*, 2019.
5. J. A. Suykens and J. Vandewalle, “**Least squares support vector machine classifiers**,” *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.
6. L. Su, L.-F. Yu, and Y.-H. Yang, “**Sparse cepstral, phase codes for guitar playing technique classification**.” in Proceedings of the *15th International Society for Music Information Retrieval Conference (ISMIR)*, 2014, pp. 9–14.
7. Y.-P. Chen, L. Su, and Y.-H. Yang, “**Electric guitar playing technique detection in real-world recording based on f0 sequence pattern recognition**.” in Proceedings of the *16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015, pp. 708–714
8. P.-C. Li, L. Su, Y.-h. Yang, A. W. Su et al., “**Analysis of expressive musical terms in violin using score-informed and expression-based audio features**.” in Proceedings of *the 16th International Society for Music Information Retrieval Conference (ISMIR), 2015*, pp. 809–815.
9. K. hoi, D. Joo, and J. Kim, “**Kapre: On-gpu audio preprocessing layers for a quick implementation of deep neural network models with keras**,” in *Machine Learning for Music Discovery Workshop at 34th International Conference on Machine Learning. ICML*, 2017
10. D. P. Kingma and J. Ba, “**Adam: A method for stochastic optimization,**” in *3rd International Conference on Learning Representations (ICLR*), 2015.
11. S. Osowski, K. Siwek, and T. Markiewicz, “**Mlp and svm networks - a comparative study**,” in Proceedings of the *6th Nordic Signal Processing Symposium, 2004. NORSIG 2004. IEEE*, 2004, pp. 37–40.
12. K. Choi, G. Fazekas, and M. B. Sandler, “**Automatic tagging using deep convolutional neural networks**,” in *17th International Society for Music Information Retrieval Conference (ISMIR)*, 2016, pp. 805–811.

13. K. Simonyan and A. Zisserman, **"Very deep convolutional networks for large-scale image recognition**," in *3rd International Conference on Learning Representations (ICLR)*, 2014.
14. K. Choi, G. Fazekas, M. Sandler, and K. Cho, "**Convolutional recurrent neural networks for music classification**," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2392–2396.
15. I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, **Deep learning**. *MIT press Cambridge*, 2016, vol. 1.
16. Hassan, Malik & Mirza, Waseem & Hussain, Mirza **Header Based Spam Filtering Using Machine Learning Approach** *International Journal of Emerging Technologies in Engineering Research (IJETER)* Volume 5, Issue 10, October (2017).