Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)
Vol. 10, No. 4, December 2024, pp. 992-1003
ISSN: 2338-3070, DOI: 10.26555/jiteki.v10i4.30526

KNN-Based Music Recommender System with Feedforward Neural Network

Andhika Loiz, Z. K. Abdurahman Baizal
School of Computing, Telkom University, Bandung 40257, Indonesia

ARTICLE INFO

Article history:

Received December 31, 29024
Revised January 23, 2025
Accepted February 01, 2025

Keywords:

Autonomous navigation;
Service robot;
Web app;
Robot operating system;
SLAM

ABSTRACT

Music, as a form of entertainment, is now an essential element in the lives of many individuals. Access to music-related information has become widespread through various websites and applications, leading to a significant increase in music data. Technological advancements have driven the development of music recommendation system research, which utilizes multiple methods, algorithms, and classification techniques to present recommendations that match user preferences. This research contributes to integrating the K-Nearest Neighbors (KNN) method for initial classification and the more advanced Feedforward Neural Network (FNN) model. In addition, this research also recommends songs with similar audio features. The main focus of this research is to design and evaluate a song recommendation system by combining such methods while comparing various hyperparameter results to find the most suitable model. The best model found will be incorporated into Content-Based Filtering (CBF) to provide song recommendations based on genre. This research uses the GTZAN dataset of 1,000 audio data from ten music genres. The K-NN model test assesses how well the model maintains consistency and achieves optimal performance. This study conducted three tests to find the best-performing model by integrating the model and hyperparameters. The results showed that the third FNN model showed the best performance after being optimized using the SGD optimizer. Furthermore, this model was combined with the CBF method using cosine similarity calculation. The system effectively recommended songs based on the blues genre, with five relevant nearest neighbors and an average score reaching 98%.

Corresponding Author:

Z. K. A. Baizal, School of Computing, Jl. Terusan Buah Batu, Bandung 40257, Indonesia
Email: baizal@telkomuniversity.ac.id

1. INTRODUCTION

Many consider music an essential form of entertainment [1]. The development of information technology and the popularity of music streaming platforms have changed how people enjoy and listen to music. People also increasingly classify music into specific categories called music genres. As a result, we now recognize a wide variety of music genres [2], [3]. Based on APJII data, around 35.5% of internet users in Indonesia, around

46.9 million people, listen to music online [4]. Music platforms such as Spotify, Apple Music, and Deezer allow people to access an unlimited variety of music and artists worldwide. However, with so many options, users often struggle to find songs that suit their tastes. For this reason, recommender systems have become very useful in helping them find suitable music [2]. Recommender systems have become very popular in the entertainment industry, especially music. In recommender systems, several methods can be used, such as Collaborative Filtering (CF), Content-Based Filtering (CBF), as well as a combination of both, known as a Hybrid [5], [6].

Research by Adiyar and friends [7] aims to overcome the "cold start" problem and promote lesser-known musicians by developing a content-based music recommender system that prioritizes acoustic similarities between songs without relying on external features such as genre or artist. The analysis showed that the more complex the method, the better the system performance. Random recommender has the lowest performance (mean precision@10 = 0.006, mean nDCG = 0.006), followed by the genre-based approach (mean precision@10 = 0.066, mean nDCG = 0.066), while acoustic analysis shows better results (mean precision@10 = 0.112, mean nDCG = 0.125). However, the weakness of this study lies in the low quality of the metrics. Research by Ahmed and friends [8] discussed various deep learning architectures for music genre classification, emphasizing the advantages of a modified Convolutional Neural Network (CNN) that achieved 92.7% accuracy on the GTZAN dataset. Feedforward Neural Network (FNN) is used as one of the classification models, along with other models such as CNN, Support Vector Machine (SVM), k-nearest Neighbors (kNN), and Long Short-term Memory (LSTM). This research also evaluates the performance of CNN against other models, such as RNN-LSTM, SVM, and kNN, to demonstrate the effectiveness of cutting-edge deep learning approaches in this field. In addition, it highlights the importance of feature selection strategies, using Mel-Frequency Cepstral Coefficients (MFCC) for feature extraction, and explores the applicability of such models in the development of music recommendation systems. However, the weakness of this research is that Convolutional Neural Networks (CNN) models are often considered difficult to understand in depth due to their "black box" characteristics and have limitations in terms of interpretation. In addition, conventional datasets such as GTZAN are considered to underrepresent the diversity of contemporary music genres. Therefore, it is necessary to consider using a broader and more diverse dataset to improve the model's generalization ability.

However, the weakness of this study is that the applied model is not always optimal in all situations. Research [9] discusses the application of CBF in music recommender systems. To improve the user experience in finding music, the CBF method is combined with deep Neural Networks, specifically CNN. This study tested four models with different configurations and achieved an accuracy of about 73.52%. Deep learning can potentially improve the performance of music recommender systems. However, this research has some weaknesses. One is the need for extensive computational resources to train deep Neural Network models, which can be an obstacle if resources are limited. In addition, the time-consuming training process is also a challenge, although the model can produce reasonably accurate predictions. Research by Hafidh and friends [1] discusses K-Nearest Neighbors (KNN) aimed at measuring user satisfaction using the System Usability Scale (SUS) and helping users find music that matches their preferences through a recommender system that considers user and music criteria. The KNN method is used to find K-Nearest Neighbors in music data based on lyrics to provide a relevant recommender. KNN has a simple and effective numerical classification algorithm. Although the accuracy results are not mentioned, this research focuses on measuring user satisfaction, reflected in the SUS score of 83.65 with a rating of A, which belongs to the excellent category.

Additional analysis may be required to evaluate accuracy in more detail. Research by Utomo and friends [10] discusses a music recommender system using the KNN algorithm to determine the best machine learning technique for predicting song genres and developing a cosine similarity-based system. The results showed that KNN had an accuracy of 64.9%, while Support Vector Machine (SVM) reached 77%. The cosine similarity-based recommender system obtained an average accuracy of 80% in recommending songs with similar genres. The study's weakness lies in the reliance on MFCC features, which may not cover all essential aspects for genre classification. Using KNN and SVM without PCA may reduce accuracy, as PCA can improve model performance. In addition, the dataset's size and diversity may limit the model's ability to be applied to more diverse genres.

Thus, research on music recommender systems that utilize various combinations of methods, algorithms, and classification techniques continues to evolve along with technological advancements. Ultimately, recommender systems play an important role in suggesting music that matches users' preferences.

This research contributes to utilizing the KNN approach for the initial classification of song genres, Feedforward Neural Networks (FNN) as an advanced model to improve accuracy in music genre classification, and CBF to recommend songs based on the similarity of audio features. The main focus is to develop and evaluate a song recommendation system by integrating these methods while comparing the results of various hyperparameters to determine the optimal model. The results include top k accuracy metrics for genre classification and relevant song recommendations. The best FNN model integrated with CBF in this study provides song recommendations by considering the similarity of audio features between existing songs and query items. Little research has integrated classification methods with recommendation systems, as done in this study. This research contributes to classifying music genres and enhancing the user experience through personalized recommendation systems by integrating the three approaches.

This research is divided into several sections and subsections. The second section reviews previous research addressing similar topics as well as aspects related to system design. The third section reviews the proposed methodology, focusing on the development of a recommendation system using Content-Based Filtering, KNN, and FNN approaches, complemented by some experiments as part of the evaluation. The fourth section discusses the system development process using the proposed method along with the final results obtained.

2. METHOD

Fig. 1 the system designed in this study. The process includes data collection, pre-processing, data division for training and testing, KNN implementation as the initial model, hyperparameter adjustment, result analysis, FNN implementation, and FNN performance evaluation using the top-k accuracy metric.

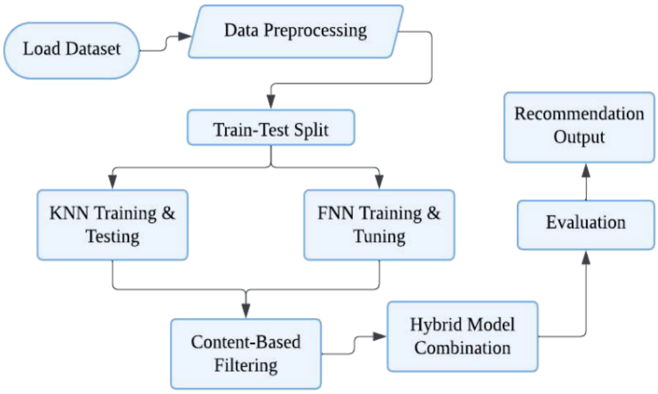

Fig. 1. KNN-based & FNN design

2.1. Data Collection

Researchers widely use the GTZAN dataset to analyze music genres and perform music genre classification [11]. This dataset includes ten music categories, namely classical, blues, alternative, hip-hop, disco, jazz, pop, metal, rock, and reggae, which can be downloaded through the Kaggle platform. Each genre consists of twenty-five 30-second mono audio recordings, with a bit depth of 16 bits and a sample rate of 22,050 Hz [9]. This dataset consists of 1,000 songs with a size of 1.6 GB [12]. Research [13] compared various algorithm models using the GTZAN dataset. Some of the models tested include Artificial Neural Network (ANN), Support Vector Machine (SVM), Convolutional Neural Network (CNN), Decision Tree, and Multilayer Perceptron (MLP). The results show that the GTZAN dataset is suitable for testing music genre classification models, with CNN achieving the highest accuracy of around 91%. Research [14] used a Decision Tree model to classify music genres based on audio data. The model achieved a training accuracy of 75% and a testing accuracy of 54%, indicating potential overfitting. The study also utilized the GTZAN dataset, emphasizing the need for further development to improve accuracy and user experience. Both studies confirmed that the GTZAN dataset is very feasible to use, thanks to the wide variety of music genres and the consistency of the number of samples in each genre.

The dataset is organized by music genre and available samples [15]. Table 1 presents an initial summary of the distribution of each genre and the number of samples it has, providing an overview of the proportions of the data. In addition, Table 2 the dataset used, the file “features_30_sec.csv”. This table structure is designed to make it easier to understand and identify patterns in the data.

Table 1. Distribution each genre and samples

No	Genre	Number of samples
1	Blues	25
2	Classical	25
3	Country	25
4	Disco	25
5	Hip hop	25
6	Jazz	25
7	Metal	25
8	Pop	25
9	Reggae	25
10	Rock	25

Table 2. Detailed of the dataset

Filename	Length	....Label
Blues.00000.wav	661794	....Blues
Blues.00001.wav	661794	....Blues
Blues.00002.wav	661794	....Blues
....	....	....
Rock.0097.wav	661794	....Rock
Rock.0098.wav	661794	.... Rock
Rock.0099.wav	661794	....Rock

2.2. Data Preprocessing

Data pre-processing is a crucial early stage in data management, especially in preparing text for analysis. This step includes a series of processes, such as cleaning and alignment. The main goal is to ensure that the data becomes cleaner, structured, and ready to be used in various further analysis techniques so that the results obtained become more accurate and reliable [16], [5], [17].

This research has removed columns that were deemed irrelevant during the dataset preparation stage. These columns are potential noise and have no relation to the target or contribution to helping the model determine the target. After that, the data is separated into feature (X) and target (y) variables. The X variable contains the feature data, while the y variable contains the label or target to be predicted. All columns in X are then converted to numeric type, with non-numeric values automatically replaced with NaN (Not a Number) if found. Rows containing NaN are removed so that the data in X only consists of valid numeric values. Table 3 represents the final clean and structured result displayed in the final feature variable.

Table 3. Result of feature variable

length	Chroma_sft_mean	....	Mfcc20_mean	Mfcc20_var
661794	0.350088	....	1.221291	46.936035
661794	0.340914	....	0.531217	45.786282
....	....	....	....	....
661794	0.362485	....	-3.590644	41.299088
661794	0.358401	....	1.155239	49.662510

The target "y" is converted to a numeric value, such as [0, 0, 1, 1, 2, 2, 3…9, 9], using LabelEncoder. Furthermore, to maintain the length balance between the feature data (X) and the target $(y\_encoded)$, adjustments are made, especially if the removal of "NaN" values in X results in a difference in the number of lines between the two. After the preprocessing process is complete, the next step in this research is to separate the dataset into training and test data. Training data is used to train the KNN model, while test data is used to evaluate the model's ability to predict data that has never been encountered before. The division is done with an allocation of 80% for training data and 20% for test data.

2.3. K-Nearest Neighbors (KNN) models

In cases where information about the data distribution is limited, the K-Nearest Neighbors (K-NN) classification method is often used as an initial approach in classification research [16]. This method can be applied to solve various problems in both regression and classification [18]. K-NN does not perform a training process before receiving data. Hence, it is known as a "lazy learning" algorithm. Nonetheless, it is quite effective in many cases. The basic principles K-NN uses also allow the development of various other implementations [19], [20].

The KNN algorithm forms the basis of our implementation logic, which is carried out through the following steps:

1. The training set A consists of elements $A = a1, a2, a3, \dots, an$ with classes $\{C = c1, c2, c3, \dots, cs\}$.
2. The main objective is to find the k nearest neighbors in the training set with the smallest distance to the elements in the test set.
3. For each element in the test set, the program iterates through all elements in the training set. At each iteration, the program calculates by utilizing the Manhattan parameter as the best parameter in KNN between each element for $1 \ll= i \ll= n$ of the set and CJ, $1 \ll= i \ll= n$ of the test set.
4. The results of distance and class calculations will be stored in a list. This list is sorted by distance, and the kk nearest neighbors are taken from the list.
5. Each neighbor in the list of k neighbors votes based on its class, and the class with the highest number of votes will be selected as the classification result.
6. For the next element in the test set, the program repeats the same process as step 3 [18]. efficie

This research applies to the K-NN algorithm optimized with GridSearchCV. Fig. 2 illustrates the KNN-based custom-designed architecture. GridSearchCV is a mathematical method to determine the optimal hyperparameter combination to improve classification model performance. This technique evaluates various pre-defined hyperparameter combinations [21]. This process is usually combined with k-fold cross-validation to ensure the best hyperparameter selection [22]. Afterward, an initial evaluation is conducted to assess the model's performance.

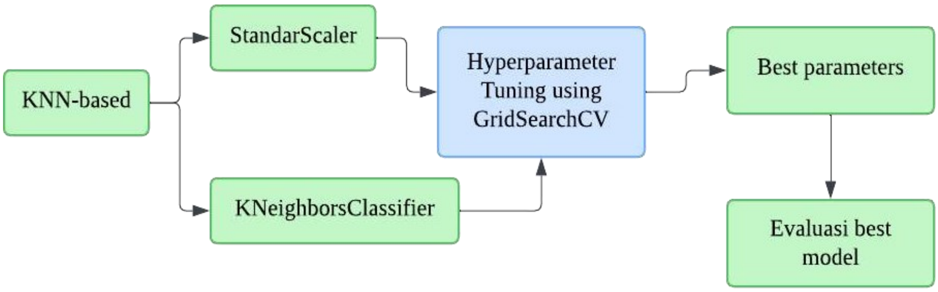

Fig. 2. Architecture KNN

This study tested a KNN-based model with five folds of 24 possible combinations, resulting in 120 fits. This process was successfully applied on a small scale. Furthermore, an evaluation was conducted to assess the performance of the KNN model, including an analysis of the relationship between genre and tempo. The model can identify genre labels based on tempo patterns in the dataset. Evaluation is done using precision, recall, and f1-score metrics.

This study divides the data into "y" and "X" variables to be normalized. Normalization is applied to align the data attributes into the same range, thus allowing a fairer comparison between attributes with different scales [23]. The normalization process uses the MinMaxScaler method, which converts each feature value into a specific range, usually between 0 and 1. This method is done by subtracting the minimum value of the feature from each value and dividing it by the range of the feature value. MinMaxScaler is a very effective method for features with a known range of values requiring uniform scale adjustment. It is ideal for machine learning algorithms that require consistency of scale between features [24], [25], [26]. However, its weakness lies in its sensitivity to outliers, as extreme values can affect the normalization scale. Once the normalization process is complete, Principal Component Analysis (PCA) is applied to optimize the two main components in the data. PCA converts the data into a new set of independent or uncorrelated components sorted by their respective contributions to the total variance in the data [27].

The KNN pipeline was designed to evaluate the stability of the data after the normalization process. Unlike the previous method, this pipeline involves 150 trials using a five-step scheme on 30 candidate data. Fig. 3 shows the structure of the KNN pipeline. Next, the model is re-evaluated to assess its performance. Before moving on to the next model, the seed value is set to prevent local convergence and bias. In addition, callbacks are applied to monitor the model's performance during the training process [28], [29].

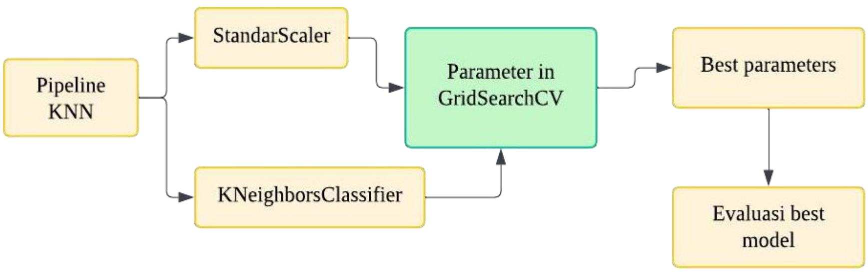

Fig. 3. Architecture K-NN Pipeline

2.4. Feed Forward Neural Network (FNN) models

Feedforward Neural Networks (FNN) are the core of the learning architecture in neural networks. FNN is a type of neural network that flows data in only one direction, from the input layer through the hidden layers, until it reaches the output layer. FNNs are designed to solve nonlinear problems. Data from the input layer is processed in the hidden layer; then, the results are forwarded to the output layer as the network output. If there is a difference between the actual and expected outputs, the calculated error will be sent back to the input layer as feedback. Furthermore, the weights in each network layer will be adjusted using the gradient descent method based on the error that occurs [30], [31], [32].

A feedforward neural network consists of L layers; each layer has n_l neurons [33]. The relationship between the $l-1$ layer and the lth layer is represented by a weight matrix $W^1 \in \mathbb{R}^{n_L \times n_{L-1}}$, with a bias vector $b^1 \in \mathbb{R}^{n1}$. The total number of neurons in the network is calculated as $N = \sum_{l=1}^{L} n_1$ while the total number of connections in the network is $E = \sum_{l=2}^{L} n_1 \cdot n_{l-1}$.

Data preparation is performed before applying the FNN, which includes separating features from the target
variable. Next, the data was normalized using StandardScaler and divided into 80% for training data and 20% for validation data. In this study, NN is utilized as a modern model to learn complex patterns in data, allowing CBF to capture non-linear relationships between features. This study uses the TensorFlow Keras library to build various NN model structures. The FNN model includes several layers, such as the input layer, convolution layer, max pooling layer, flatten layer, dense layer, dropout layer, and output layer [5], [33], [34].

This study developed four FNN models to identify the model with the best prediction accuracy. The models only utilize dense layers with the number of units adjusted to the needs of each model (512, 256, 128, 64, 10) without using all common elements in the FNN structure. To minimize the risk of overfitting, some models are equipped with a dropout of 0.2 or 0.3 as needed [34]. Various optimization algorithms, such as the Adam model, SGD, and RMSprop, are applied to obtain the optimal layer configuration [35], [36]. The evaluation was performed using the Top-k accuracy metric [37], which describes the set of k classes with the highest probability predicted by the model. Each class has a confidence value representing its predictive probability [38], [39]. The cross-validation technique assesses the model's performance and generalization ability. This method also reduces the possibility of overfitting if the data is only shared once [40].

2.5. Content Based Filtering (CBF)

Content filtering-based recommender systems analyze the content of items previously used or liked by users and then recommend other similar items in line with the user's preferences and interests. In addition, this system is also able to identify items that have similarities with items that have been recognized or approved by users [41], [42], [43]. This method is relevant to this research because it aims to generate recommender based on the similarity between question items. The cosine similarity between the query and all other items is calculated using a TF-IDF vectorizer as the first step in applying the content-based filtration method. The calculation results generate recommender based on itemindices and filenames, sorted according to the similarity score [44], [45]. In CBF, a 'search based' is obtained from the query index, so CBF is recommended based on the selected index.

3. RESULTS AND ANALYSIS

3.1. K-NN and pipeline K-NN result

The KNN-based model aims to evaluate how much the model can achieve optimal performance. The KNN-based algorithm performs classification by calculating the distance between samples. At the same time, feature standardization is applied using StandardScaler to ensure that the data has a mean of 0 and a variance of 1. In this study, the number of neighbors considered is [3, 5, 7, 9], with two types of weighting: "uniform" (all neighbors have the same weight) and 'distance' (weights are determined based on distance) [46], [47]. Distance measurement between samples is performed using Euclidean, Manhattan, and Minkowski. The results of hyperparameter tuning with GridSearchCV showed that the best configuration was a model with three nearest neighbors, weighting by distance and the "Manhattan" distance metric. Table 4 shows the final evaluation of the model, which was performed using accuracy, precision, recall, and F1-score metrics [48].

Table 4. KNN-based results

	Precision	recall	F1-score
blues	0.72	0.65	0.68
classical	0.92	0.85	0.88
country	0.75	0.67	0.71
disco	0.61	0.67	0.64
hiphop	0.62	0.87	0.72
jazz	0.77	0.77	0.77
metal	0.96	0.88	0.92
pop	0.82	0.69	0.75
reggae	0.60	0.52	0.56
rock	0.50	0.62	0.55

The model achieved an accuracy rate of 71% with an average F1-Score of 72%. Some classes performed very well with an F1-Score above 0.88, while others performed less well with an F1-Score below 0.60. Previous research examined the application of the K-Nearest Neighbors (KNN) algorithm in music genre classification, focusing on comparing execution time and accuracy rate. The tests were conducted using the GTZAN dataset. Although the execution time was drastically reduced from about 80 seconds to 1 second, the preprocessing method caused a significant decrease in classification accuracy, from about 70% to less than 12% [20]. Before normalization, the initial stage involved analyzing the wave patterns of the songs to determine genre labels using the BPM (Beats Per Minute) feature. To ensure consistency of the results, the data was then normalized. Fig. 4 shows the results of PCA analysis for the genre after normalization, which shows that data distribution is maintained. Thus, the KNN pipeline was built based on the KNN evaluation, confirming that the data remained stable after normalization.

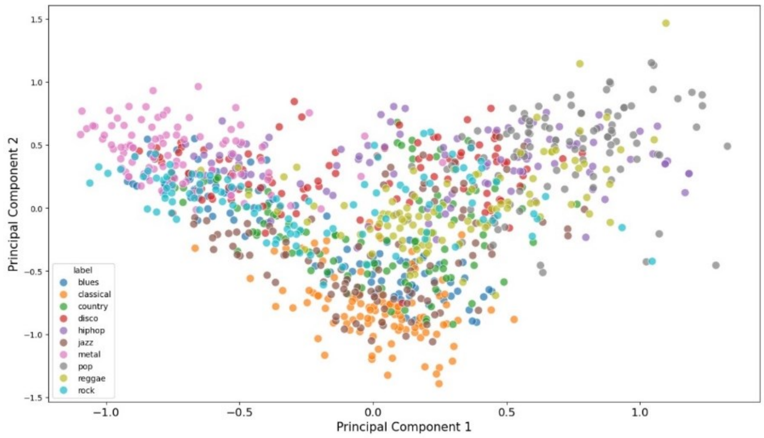

Fig. 4. PCA on genre

The KNN pipeline, just like the KNN-based model, combines standardization with the KNN-based model to classify the data optimally. However, after the data was normalized, this pipeline was applied, and evaluation results were successfully maintained, equivalent to KNN-based, as seen in Table 4. In addition, the pipeline also showed consistent performance, indicating that the model is reliable and not affected by interference.

3.2. FNN result

FNN is applied to perform classification. The data was processed through a preprocessing stage, including removing irrelevant columns and using label encoding. After that, the data was normalized and divided into 80% for training and 20% for testing. This study analyzed four models with varying numbers of dense layers and dropout rates to identify the most optimal model. Model performance was assessed based on accuracy, with higher accuracy values indicating better model performance [9]. The first model consists of dense layers without dropouts, with 256, 128, 64, and len units. The activation functions used were fundamental for the hidden layer and softmax for the output layer. The second model adds a dropout of 0.2 to each dense layer while still using the relu activation function in the hidden layer and softmax in the output layer. Both models use the Adam optimizer. The third model uses the SGD optimizer with the same configuration of the number of dense units and dropout levels as in the second model. In contrast, the fourth model uses the RMSprop optimizer, increasing the number of dense units to 1024, 512, 256, 128, 64, and len and increasing the dropout rate from 0.2 to 0.3 in each dense layer. Table 5 displays the evaluation results of the four models using the Top-k accuracy metric.

Table 5. Overall model results

	Top 5 accuracy (%) Adam	Top 5 accuracy (%) SGD	Top 5 accuracy (%) RMSprop
Model 1	0.9650	-	-
Model 2	0.9700	-	-
Model 3	-	0.9800	-
Model 4	-	-	0.9500

Table 5 reveals that Model 3, which uses the SGD optimizer, achieved the best results in this test. The combination of a dropout rate of 0.2 with the SGD optimizer proved to be very effective. The superior performance of Model 3 shows that the SGD optimizer can produce better generalization than other optimizers with a top five accuracy of 0.98, especially with architectures with moderate dropout rates. K-NN and FNN models are used to evaluate the best performance through optimization. Previous research explored various deep-learning architectures for music genre classification tasks. This research focuses on the advantages of a modified Convolutional Neural Network (CNN), which achieved 92.7% accuracy on the GTZAN dataset. Feedforward Neural Network (FNN) is one of the classification models applied in this research, along with other models such as Support Vector Machine (SVM), K-Nearest Neighbors (K-NN), and Long Short-Term Memory (LSTM) [31].

This model pipeline consists of K-NN and FNN, designed to predict the genre of songs in a query. The FNN model uses the best NN model, the third model, with SGD optimization as its input. After that, the pipeline is evaluated to assess its performance. If there is a difference in prediction results between K-NN and FNN, the system will choose one of the genres that best match the results from K-NN or FNN.

3.3. Recommender result

The system recommends items based on the queries asked using this section's Content-Based Filtering (CBF) method. The data is filtered by genre, and the cosine similarity method determines the degree of similarity between a query item and another item. Cosine similarity measures the cosine angle between two objects and compares two files on a normalized scale. The calculation is done by calculating the dot product of the two identity vectors [49], [50]. Once the similarity score is obtained, the query items and all items are weighted and evaluated using accuracy, recall, and Mean Average Precision (MAP) metrics. MAP and recall are the main metrics to assess the relevance of recommendations. In this evaluation, the model performed with an accuracy of 0.77, MAP of 0.85, and recall of 0.98. Previous research discussed the application of Content-Based Filtering (CBF) in music recommendation systems. This method was combined with a deep Neural Network (DNN), specifically CNN, to improve the user experience when searching for music. The study tested four different model configurations and recorded an accuracy of 73.52%. This result indicates that deep learning can improve the performance of music recommendation systems [9]. Fig. 5 illustrates the performance improvement of the model.

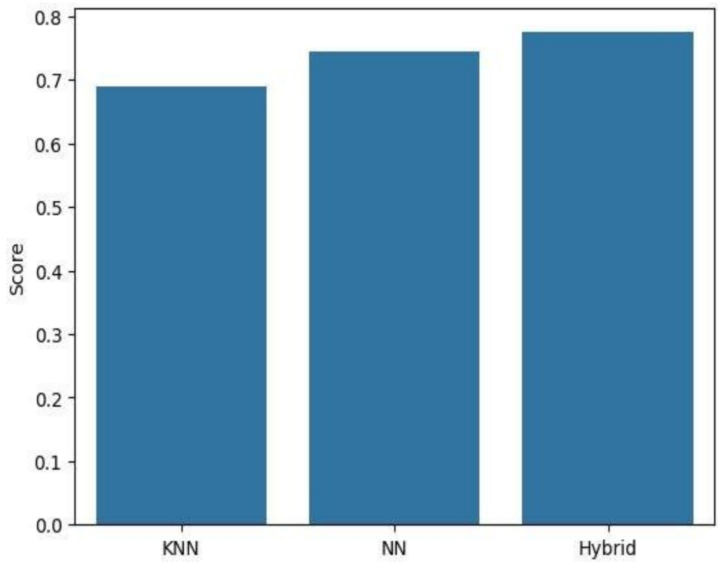

Fig. 5. Comparison of model accuracies

Fig. 5 shows that the hybrid model combining K-NN and FNN improves the performance in providing music recommendations according to user preferences or content-based filtering methods. Furthermore, based on the query given by the hybrid model, the system presents recommendations based on index, score, and genre. Table 6 displays the recommendations' results. Table 6 is organized based on the song's query features and calculates its similarity to other songs in the dataset. The five songs with the highest degree of similarity will be recommended.

Table 6. Recommenders result

Hybrid Recommended Indices	Hybrid Recommended Scores	Hybrid Recommended Genres
36	0.99987453	‘blues’
45	0.99975488	‘blues’
31	0.99972421	‘blues’
6	0.99970859	‘blues’
42	0.99894372	‘blues’

4. CONCLUSION

This research integrates CBF, KNN, and FNN methods to develop an effective music recommender system. The system utilizes KNN to perform the initial classification of music genres. At the same time, FNN is used to identify non-linear patterns in the audio data, thus improving the system's overall performance. The KNN model achieved an accuracy of 71% with an average F1 score of 72%. On the other hand, the FNN model, trained using the Stochastic Gradient Descent (SGD) algorithm, achieved the highest Top 5 accuracy of 98%. CBF-based song recommendations with cosine similarity have also proven effective in providing relevant music suggestions based on the similarity of acoustic features evaluated by accuracy, recall, and MAP. Thus, the results obtained from CBF show an accuracy of 0.77, MAP 0.85, and recall of 0.98. Recommendations are obtained based on recommender indices, scores, and genres, giving five indices similar to the blues genre with an average score of 98%.

This research has limitations, such as high computational resource requirements and less stable performance in specific music genres. Nonetheless, this research makes a meaningful contribution to developing a more personalized and adaptive music recommendation system. To improve the scalability and performance of the system, it is recommended that future research adopt a hybrid approach and address challenges related to cold start issues. Based on the KNN and NN model testing analysis, the system is highly dependent on extensive data, making it sensitive to adding new items that do not have complete feature descriptions. However, it should be noted that the GTZAN dataset may contain biases due to the uneven distribution of genres, where some genres are more dominant than others. This may affect the performance of the model, especially in underrepresented genres. The limitation of 30 seconds of duration per song may also affect the model's ability to generalize to music with more varied durations. Therefore, Future research is recommended to expand the dataset with more diverse genres and representative data from different musical cultures to increase variation and reduce bias.

New users who have no initial preference data or history of songs listened to may also experience the same problem. Although more commonly found in Collaborative Filtering (CF), the cold start problem can also occur in CBF. This research analyzes acoustic features by utilizing 150 CBF features described in section 2.3. Since not all songs have complete features, the cold start problem on new items is not significant. However, the main problem remains with new users who do not have initial preferences. Thus, combining various machine learning techniques has an excellent opportunity to create a more precise, relevant, and personalized music recommender system. By continuing to address the challenges, this approach can bring innovations to the listening experience, provide a recommender better aligned with user preferences, and improve overall satisfaction.

REFERENCES

[1] M. H. Nofal, Z. K. A. Baizal, and R. Dharayani, “Multi Criteria Recommender System for Music using K-Nearest Neighbors and Weighted Product Method,” *Indonesia Journal on Computing*, vol. 6, no. 2, pp. 33–42, 2021, https://doi.org/10.34818/indojc.2021.6.2.575.

[2] W. Yun, L. Jian, and M. Yanlong, “A Hybrid Music Recommendation Model Based on Personalized Measurement and Game Theory,” *Chinese Journal of Electronics*, vol. 32, no. 6, pp. 1319–1328, 2023, https://doi.org/10.23919/cje.2021.00.172.

[3] A. Budhrani, A. Patel, and S. Ribadiya, “Music2Vec: Music Genre Classification and Recommendation System,” *Proceedings of the 4th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2020*, pp. 1406–1411, 2020, https://doi.org/10.1109/ICECA49313.2020.9297559.

[4] J. Sihombing, A. Tuhagana, and D. Triadinda, “The Effect of Promotion and Hedonic Shopping On Impulse Buying On Spotify Applications,” *Neo Journal of economy and social humanities*, vol. 1, no. 3, pp. 194–204, 2022, https://doi.org/10.56403/nejesh.v1i3.47.

[5] A. Nilla and E. B. Setiawan, “Film Recommendation System Using Content-Based Filtering and the Convolutional Neural Network (CNN) Classification Methods,” *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 10, no. 1, p. 17, 2024, https://doi.org/10.26555/jiteki.v9i4.28113.

[6] A. Bin Suhaim and J. Berri, “Context-Aware Recommender Systems for Social Networks: Review, Challenges and Opportunities,” *IEEE Access*, vol. 9, pp. 57440–57463, 2021, https://doi.org/10.1109/ACCESS.2021.3072165.

[7] A. Niyazov, E. Mikhailova, and O. Egorova, "Content-based Music Recommendation System," *Proceeding Of The 29th Conference Of Fruct Association*, pp. 274-279, 2021, https://doi.org/ 10.23919/FRUCT52173.2021.9435533.

[8] M. Ahmed, U. Rozario, M. M. Kabir, Z. Aung, J. Shin, and M. F. Mridha, "Musical Genre Classification using Advanced Audio Analysis and Deep Learning Techniques," *IEEE Open Journal of the Computer Society*, vol. PP, pp. 1–12, 2024, https://doi.org/10.1109/OJCS.2024.3431229.

[9] R. Andiety and Z. K. A. Baizal, "Content-Based Music Recommender System Using Deep Neural Network," *Technology and Science (BITS)*, vol. 6, no. 2, pp. 1111–1119, 2024, https://doi.org/10.47065/bits.v6i2.5762.

[10] B. Sreedhar *et al.*, "Music Recommendation System Based on Facial Emotion," *Journal of Humanities,Music and Dance*, vol. 3, no. 3, 2023, https://doi.org/10.55529/jhmd.33.11.22 .

[11] R. Tang, M. Qi, and N. Wang, "Music style classification by jointly using CNN and Transformer," *ACM International Conference Proceeding Series*, pp. 707–712, 2024, https://doi.org/10.1145/3651671.3651696.

[12] Y. H. Cheng, P. C. Chang, and C. N. Kuo, "Convolutional neural networks approach for music genre classification," *Proceedings - 2020 International Symposium on Computer, Consumer and Control, IS3C 2020*, pp. 399–403, 2020, https://doi.org/10.1109/IS3C50286.2020.00109.

[13] N. Ndou, R. Ajoodha, and A. Jadhav, "Music genre classification: A review of deep-learning and traditional machine-learning approaches," *2021 IEEE International IOT, Electronics and Mechatronics Conference, IEMTRONICS 2021 - Proceedings*, pp. 1-6, 2021, https://doi.org/10.1109/IEMTRONICS52119.2021.9422487.

[14] L. Novea and B. H. Prasetio, "Sistem Prediksi Genre Musik dan Penyediaan Tautan Rekomendasi Daftar Putar Menggunakan Teknik STFT dan Decision Tree Machine Learning," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 1, no. 1, pp. 1–15, 2020, [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/13958.

[15] A. Ghildiyal, K. Singh, and S. Sharma, "Music Genre Classification using Machine Learning," *Proceedings of the 4th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2020*, pp. 1368–1372, 2020, https://doi.org/10.1109/ICECA49313.2020.9297444.

[16] H. T. Duong and T. A. Nguyen-Thi, "A review: preprocessing techniques and data augmentation for sentiment analysis," *Comput Soc Netw*, vol. 8, no. 1, pp. 1–16, 2021, https://doi.org/10.1186/s40649-020-00080- x.

[17] Y. Zhang, K. Wu, and M. Zhao, "An Audio-Visual Separation Model Integrating Dual-Channel Attention Mechanism," *IEEE Access*, vol. 11, no. June, pp. 63069–63080, 2023, https://doi.org/10.1109/ACCESS.2023.3287860.

[18] M. Ahmed, U. Rozario, M. M. Kabir, Z. Aung, J. Shin, and M. F. Mridha, "Musical Genre Classification using Advanced Audio Analysis and Deep Learning Techniques," *IEEE Open Journal of the Computer Society*, vol. 5, no. July, pp. 457–467, 2024, https://doi.org/10.1109/OJCS.2024.3431229.

[19] L. Xiong and Y. Yao, "Study on an adaptive thermal comfort model with K-nearest-neighbors (KNN) algorithm," *Build Environ*, vol. 202, no. December 2020, p. 108026, 2021, https://doi.org/10.1016/j.buildenv.2021.108026.

[20] X. Mu, "Implementation of Music Genre Classifier Using KNN Algorithm," *Highlights in Science, Engineering and Technology*, vol. 34, pp. 149–154, 2023, https://doi.org/10.54097/hset.v34i.5439.

[21] G. N. Ahmad, H. Fatima, Shafiullah, A. Salah Saidi, and Imdadullah, "Efficient Medical Diagnosis of Human Heart Diseases Using Machine Learning Techniques with and Without GridSearchCV," *IEEE Access*, vol. 10, pp. 80151–80173, 2022, https://doi.org/10.1109/ACCESS.2022.3165792.

[22] K. H. Liland, J. Skogholt, and U. G. Indahl, "A New Formula for Faster Computation of the K-Fold Cross-Validation and Good Regularisation Parameter Values in Ridge Regression," *IEEE Access*, vol. 12, pp. 17349–17368, 2024, https://doi.org/10.1109/ACCESS.2024.3357097.

[23] A. Toha, P. Purwono, and W. Gata, "Model Prediksi Kualitas Udara dengan Support Vector Machines dengan Optimasi Hyperparameter GridSearch CV," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 4, no. 1, pp. 12–21, 2022, https://doi.org/10.12928/biste.v4i1.6079.

[24] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," *Appl Soft Comput*, vol. 97, p. 105524, 2020, https://doi.org/10.1016/j.asoc.2019.105524.

[25] M. and H.-M. T. Cao-Van, Kien and Le-Gia, Minh and Tran-Cao, "Prediction of heart Failure Using Voting Ensemble Learning Models and Novel Data Normalization Techniques," *Social Science Research Network (SSRN)*, 2024, https://doi.org/10.2139/ssrn.5022046.

[26] N. Darbeheshti and E. Moradi, "LSTM-Based Forecasting Model for GRACE Accelerometer Data," *arXiv preprint arXiv:2308.0862*, 2023, [Online]. Available: http://arxiv.org/abs/2308.08621.

[27] E. Elgeldawi, A. Sayed, A. R. Galal, and A. M. Zaki, "Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis," *Informatics*, vol. 8, no. 4, pp. 1–21, 2021, https://doi.org/10.3390/informatics8040079.

[28] M. Mechó-García *et al.*, "Statistical Model of Ocular Wavefronts With Accommodation," *Invest Ophthalmol Vis Sci*, vol. 65, no. 12, p. 12, 2024, https://doi.org/10.1167/iovs.65.12.12.

[29] I. H. Bashier, M. Mosa, and S. F. Babikir, "Sesame Seed Disease Detection Using Image Classification," *Proceedings of: 2020 International Conference on Computer, Control, Electrical, and Electronics Engineering, ICCCEEE 2020*, pp. 1–5, 2021, https://doi.org/10.1109/ICCCEEE49695.2021.9429640.

[30] M. Alkanan and Y. Gulzar, "Enhanced corn seed disease classification: leveraging MobileNetV2 with feature augmentation and transfer learning," *Front Appl Math Stat*, vol. 9, 2023, https://doi.org/10.3389/fams.2023.1320177.

[31] R. Chatterjee, R. Mukherjee, P. K. Roy, and D. K. Pradhan, "Chaotic oppositional-based whale optimization to train

a feed forward neural network," *Soft comput*, vol. 26, no. 22, pp. 12421–12443, 2022, https://doi.org/10.1007/s00500-022-07141-5.

[32] K. Yamazaki, V. K. Vo-Ho, D. Bulsara, and N. Le, "Spiking Neural Networks and Their Applications: A Review," *Brain Sci*, vol. 12, no. 7, pp. 1–30, 2022, https://doi.org/10.3390/brainsci12070863.

[33] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific Machine Learning Through Physics–Informed Neural Networks: Where we are and What's Next," *J Sci Comput*, vol. 92, no. 3, pp. 1–62, 2022, https://doi.org/10.1007/s10915-022-01939-z.

[34] I. Dhall, S. Vashisth, G. Aggarwal, "Automated hand gesture recognition using a deep convolutional neural network model," In *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 811-816, 2020, https://doi.org/10.1109/Confluence47617.2020.9057853.

[35] A. Abdullah and E. B. Setiawan, "Performance of Deep Feed-Forward Neural Network Algorithm Based on Content-Based Filtering Approach," *INTENSIF: Jurnal Ilmiah Penelitian dan Penerapan Teknologi Sistem Informasi*, vol. 8, no. 2, pp. 278–294, 2024, https://doi.org/10.29407/intensif.v9i1.22999.

[36] Y. Xue, Y. Tong, and F. Neri, "An ensemble of differential evolution and Adam for training feed- forward neural networks," *Inf Sci (N Y)*, vol. 608, pp. 453–471, 2022, https://doi.org/10.1016/j.ins.2022.06.036.

[37] W. Ramadhanti and E. B. Setiawan, "Topic Detection on Twitter Using Deep Learning Method with Feature Expansion GloVe," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 9, no. 3, pp. 780–792, 2023, https://doi.org/10.26555/jiteki.v9i3.26736.

[38] Y. Ji, Y. Cao, and J. Liu, "Pruning Large Language Models via Accuracy Predictor," *arXiv preprint arXiv:2309.09507*, pp. 1–6, 2023, [Online]. Available: http://arxiv.org/abs/2309.09507.

[39] J. Resin, "From Classification Accuracy to Proper Scoring Rules: Elicitability of Probabilistic Top List Predictions," *Journal of Machine Learning Research*, vol. 24, no. 173, pp. 1–21, 2023, https://www.jmlr.org/papers/v24/23-0106.html.

[40] C. C. Chang, C. H. Chen, J. G. Hsieh, and J. H. Jeng, "Iterated cross validation method for prediction of survival in diffuse large B-cell lymphoma for small size dataset," *Sci Rep*, vol. 13, no. 1, pp. 1–10, 2023, https://doi.org/10.1038/s41598-023-28394-6.

[41] C. Channarong, C. Paosirikul, S. Maneeroj, and A. Takasu, "HybridBERT4Rec: A Hybrid (Content- Based Filtering and Collaborative Filtering) Recommender System Based on BERT," *IEEE Access*, vol. 10, pp. 56193–56206, 2022, https://doi.org/10.1109/ACCESS.2022.3177610.

[42] I. Saifudin and T. Widiyaningtyas, "Systematic Literature Review on Recommender System: Approach, Problem, Evaluation Techniques, Datasets," *IEEE Access*, vol. 12, pp. 19827– 19847, 2024, https://doi.org/10.1109/ACCESS.2024.3359274.

[43] Y. Chen and J. Huang, "Effective Content Recommendation in New Media: Leveraging Algorithmic Approaches," *IEEE Access*, vol. 12, pp. 90561–90570, 2024, https://doi.org/10.1109/ACCESS.2024.3421566.

[44] D. Baxter *et al.*, "Recommended conventions for reporting results from direct dark matter searches," *European Physical Journal C*, vol. 81, no. 10, pp. 1–19, 2021, https://doi.org/10.1140/epjc/s10052-021-09655- y.

[45] H. Liu, X. Chen, and X. Liu, "A Study of the Application of Weight Distributing Method Combining Sentiment Dictionary and TF-IDF for Text Sentiment Analysis," *IEEE Access*, vol. 10, pp. 32280– 32289, 2022, https://doi.org/10.1109/ACCESS.2022.3160172.

[46] A. Chronopoulou, M. E. Peters, A. Fraser, and J. Dodge, "AdapterSoup: Weight Averaging to Improve Generalization of Pretrained Language Models," *EACL 2023 - 17th Conference of the European Chapter of the Association for Computational Linguistics, Findings of EACL 2023*, pp. 2009–2018, 2023, https://doi.org/10.18653/v1/2023.findings-eacl.153.

[47] H. Bei, Y. Mao, W. Wang, and X. Zhang, "Fuzzy Clustering Method Based on Improved Weighted Distance," *Math Probl Eng*, vol. 2021, no. 1, p. 6687202, 2021, https://doi.org/10.1155/2021/6687202.

[48] R. Yacouby and D. Axman, "Probabilistic Extension of Precision, Recall, and F1 Score for More Thorough Evaluation of Classification Models," *In Proceedings of the first workshop on evaluation and comparison of NLP systems*, pp. 79–91, 2020, https://doi.org/10.18653/v1/2020.eval4nlp-1.9.

[49] R. H. Singh, S. Maurya, T. Tripathi, T. Narula, and G. Srivastav, "Movie Recommendation System using Cosine Similarity and KNN," *Int J Eng Adv Technol*, vol. 9, no. 5, pp. 556–559, 2020, https://doi.org/10.35940/ijeat.e9666.069520.

[50] N. Pourmoazemi and S. Maleki, "A music recommender system based on compact convolutional transformers," *Expert Syst Appl*, vol. 255, p. 124473, 2024, https://doi.org/10.1016/j.eswa.2024.124473.

BIOGRAPHY OF AUTHORS

Andhika Loiz, is an undergraduated student of Faculty of Informatics at Telkom University, Bandung, Indonesia. Email: loizandhika@student.telkomuniversity.ac.id.

Z. K. A. Baizal, is a Dean of at the Faculty of Informatics, Telkom University, Bandung. He is currently a doctor with research interests in Recommender System, Artificial Intelligence, Text mining, and Ontology. Email: baizal@telkomuniversity.ac.id.