Research Article
Journal of Intelligient Systems: 7(2) (2024) 56-65
DOI: 10.38016/jista.1401095

# Optimization of LightGBM for Song Suggestion Based on Users’ Preferences

Ömer Mintemur[1*]

[1] Department of Software Engineering, Ankara Yıldırım Beyazıt University, Ankara, Türkiye

omermintemur@aybu.edu.tr

**Abstract**
Undoubtedly, music possesses the transformative ability to instantly influence an individual's mood. In the era of the incessant flow of substantial data, novel music compositions surface on an hourly basis. It is impossible to know for an individual whether he/she will like the song or not before listening. Moreover, an individual cannot keep up with this flow. However, with the help of Machine Learning (ML) techniques, this process can be eased. In this study, a novel dataset is presented, and song suggestion problem was treated as a binary classification problem. Unlike other datasets, the presented dataset is solely based on users' preferences, indicating the likeness of a song as specified by the user. The LightGBM algorithm, along with two other ML algorithms, Extra Tree and Random Forest, is selected for comparison. These algorithms were optimized using three swarm-based optimization algorithms: Grey Wolf, Whale, and Particle Swarm optimizers. Results indicated that the attributes of the new dataset effectively discriminated the likeness of songs. Furthermore, the LightGBM algorithm demonstrated superior performance compared to the other ML algorithms employed in this study.

**Keywords:** LightGBM, Machine Learning, Classification, Swarm Based Optimization

## Kullanıcı Tercihlerine Göre Şarkı Önerisi için LightGBM'nin Optimizasyonu

**Öz**
Müzik parçaları kesinlikle bireyin ruh halini anında etkileyebilecek dönüştürücü bir yeteneğe sahiptir. Günümüzde, büyük veri kesintisiz bir akış hızına sahiptir ve her saat yeni müzik parçaları üretilmektedir. Bir şarkının beğenilip beğenilemeyeceğini dinlemeden karar vermek kişi için çok zordur. Ayrıca müzik parçalarının üretim hızına yetişmek mümkün değildir. Ancak bu zor durum Makine Öğrenmesi yöntemleri kullanılarak kolaylaştırılabilir. Bu çalışmada, yeni bir veri seti sunulmuş ve şarkı önerisi problemi bir sınıflandırma problemi olarak ele alınmıştır. Diğer veri setlerinin aksine bu veri seti tamamen kullanıcılarının dinlendikleri şarkıyı beğenip beğenmemelerini dikkate alarak oluşturulmuştur. Makine Öğrenmesi algoritması olarak LightGBM kullanılmıştır ve bu algoritma Extra Tree and Random Forest algoritmalarıyla karşılaştırılmıştır. Bu algoritmalar üç tane sürü tabanlı optimizasyon algoritması (Grey Wolf, Whale ve Particle Swarm) ile optimize edilmiştir. Sonuçlar, yeni veri setinin öz niteliklerinin şarkının beğeni durumunu ayırt etmede başarılı olduğunu ortaya koymaktadır. Dahası, sonuçlar göz önüne alındığında, LightGBM algoritmasının diğer iki algoritmaya göre daha yüksek bir performans sergilediği gözlemlenmiştir.

**Anahtar Kelimeler:** LightGBM, Makine Öğrenmesi, Sınıflandırma, Sürü Tabanlı Optimizasyon.

## 1. Introduction

If one seeks a truly universal element in our world, it becomes readily apparent in the form of music. The influence of rhythm on human experience dates back to ancient civilizations, with notable figures such as the Egyptians, Pythagoras, and Plato recognizing its profound effects (Gentili et al., 2023; Hawkins, 2022). Recent scientific studies align with the perspectives of those venerable philosophers, providing further evidence for the universal impact of music on human beings (Bartolomeo, 2022; Loukas et al., 2022). During ancient times, the procurement of specific musical compositions posed considerable challenges. However, owing to advancements in civilizations and technology, individuals now have the unprecedented ability to access an infinite array of musical pieces instantaneously.

* Corresponding Author
E-Mail: omermintemur@aybu.edu.tr

Recieved : 7 Dec 2023
Revision : 6 Mar 2024
Accepted : 7 Apr 2024

Hence, individuals can select musical pieces based on their specific needs. However, recognizing music genres that align with our personal preferences is challenged with the vast array of musical choices. Individuals often gravitate towards a particular music genre, demonstrating a tendency to overlook other genres. Consequently, it becomes challenging to explore music that may be appreciated from diverse musical genres. This difficulty can be overcome by leveraging advancements in one of the modern fields of our time. Artificial Intelligence (AI), prevalent in our era, holds sway across all facets of our existence (Păvăloaia and Necula, 2023; Risse, 2023). As in various domains, Machine Learning (ML), regarded as one of the sub-branches of AI, can be employed for predicting musical preferences. Indeed, studies on music recommendation and genre classification using ML have witnessed widespread adoption in recent years (Farajzadeh et al., 2023; Zhao et al., 2023).

Research in this domain appears to demonstrate a prevailing focus in a specific direction. Generally, studies are engaged in music recommendation methodologies grounded in genres, which can be perceived as a form of music genre classification. A music recommendation system proposed in (Liu et al., 2023). The authors highlighted the necessity of incorporating the emotional state of a listener and augmented their ML framework accordingly. The outcomes indicated a significant enhancement in performance when the emotional state was integrated into the framework. A study employing Deep Learning (DL) algorithms and Transfer Learning (TL) (Prabhakar and Lee, 2023) introduced a music recommendation system. The authors evaluated their approach on three distinct datasets and attained state-of-the-art results across all three datasets. While the features of a music piece are typically represented in vector format, it is possible to extract a feature set tailored for Convolutional Neural Networks (CNNs) (Li et al., 2021). Such a study utilized CNNs to classify music genres (Soekarta et al., 2023). The authors utilized the GTZAN dataset and applied Mel-Frequency Cepstral Coefficients (MFCC) (Logan, 2000) to extract features specifically tailored for CNNs. The obtained results demonstrated that the authors achieved a commendable accuracy in the classification of music genres. Owing to the inherent flexibility of ML and DL algorithms, facile modifications can be implemented. A different study conducted by (Wen et al., 2024) utilized CNNs for music genre classification based on GTZAN dataset. The study proposed a novel dual attention mechanism integrated into the CNN architecture. The method yielded the accuracy of 91.4%. Another study that utilized the GTZAN dataset conducted extensive experiments on eight different ML algorithms to classify music genres (Yılmaz et al., 2022). The researchers reported that the best-performing algorithm was XGBoost, achieving an accuracy of 91.80%.

Similar to present study in the aspect of optimization, the researchers used Extra Tree (ET) ML algorithm with a hyperparameter optimization technique to classify music genres. The result suggested that ET achieved an accuracy of 92.3% (HIZLISOY et al., 2023). The authors in (Wijaya and Muslikh, 2024) employed an advanced DL algorithm known as Long Short-Term Memory (LSTM). They utilized the GTZAN and ISMIR2004 datasets, achieving an accuracy of 93.10% for GTZAN and 93.69% for ISMIR2004 datasets using LSTM. A similar study to (Soekarta et al., 2023) can be found in (Singh and Biswas, 2023). The authors mentioned about the hardness of design choices of CNNs and approached this choice problem as an optimization problem and used Genetic Algorithm (GA) to optimize the CNNs architecture. The experiments conducted on three distinct datasets revealed that CNNs designed using a GA yielded superior results compared to CNNs architectures devised through manual design. Recent music streaming platforms such as Spotify also provides vast amounts of datasets that can be achieved publicly to improve AI usage in music industry. Authors in (Yuwono et al., 2023) used publicly available dataset scraped from Spotify to classify music genres. Authors used Support Vector Machine (SVM) (Noble, 2006) for their experiments and achieved the accuracy of around 80%. To enhance the comprehensibility of the literature review, Table 1 provides an overview of the methodologies and datasets employed across the reviewed studies.

AI has undeniably demonstrated its utility in the music industry. Nevertheless, a common trend observed in the literature is the predominant focus on classifying music genres, a practice that may pose challenges in certain respects. One notable challenge arises from the dynamic nature of individuals' music genre preferences, which may evolve at different stages of their life. Another challenge emerges from the standpoint of ML and DL algorithms. In the context of recommender systems, it is imperative for the system to exhibit speed and optimization to ensure efficient and timely delivery of music recommendations. The design of networks for DL approaches is recognized as a challenging task, particularly when automated optimization algorithms are employed. This process demands substantial computational resources to achieve effective model architectures. From the perspective of ML, the utilization of optimization techniques can prove to be more beneficial, expediting the overall process. Hence, the combination of an appropriate ML algorithm and advanced optimization techniques holds the potential to create more robust recommendation systems in the music industry. Addressing the challenge of individual music preferences could be furthered by leveraging an original dataset tailored specifically to this requirement.

With these drawbacks and potential improvements in consideration, this study suggests enhancements for both a more specialized dataset tailored for music

**Table 1.** Latest Studies in the Area of Music-Genre Classification

| Study | Dataset | Method | Purpose |
|---|---|---|---|
| (Soekarta et al., 2023) | GTZAN | CNN | Genre Classification |
| (Wen et al., 2024) | GTZAN | CNN | Genre Classification |
| (YILMAZ et al., 2022) | GTZAN | XGBoost | Genre Classification |
| (HIZLISOY et al., 2023) | GTZAN | Extra Tree | Genre Classification |
| (Wijaya and Muslikh, 2024) | GTZAN & ISMIR2004 | LSTM | Genre Classification |
| (Yuwono et al., 2023) | Spotify | SVM | Genre Classification |
| This study | Newly Curated Spotify Dataset | LightGBM | Music Recommendation |

recommendation systems and a potent ML algorithm, amenable to seamless optimization through state-of-the-art optimization algorithms.

For the dataset, a more specialized collection of data sourced from Spotify. The dataset was meticulously curated, centering on users' preferences and, notably, emphasizing liked songs.

Consequently, the proposed study diverges from traditional music recommendation systems, which rely on genre categorization, instead opting to tailor recommendations based on users' individual preferences.

The newly acquired dataset manifested an issue of data imbalance. In order to address this challenge and fortify the robustness of the ML framework, the Synthetic Minority Over-sampling Technique (SMOTE) was employed to rebalance the dataset.

For the ML algorithm, necessitating both speed and reliability in terms of accuracy, LightGBM (Ke et al., 2017) was chosen in the experiments. Furthermore, the algorithm was compared to two other algorithms with similar working mechanisms as LightGBM, namely Random Forest (RF) (Ho, 1995) and Extra Tree (ET) (Geurts et al., 2006). The LightGBM itself is characterized by a high degree of hyperparameter intensity, and the majority of these hyperparameters span a range of continuous values, posing a challenge for manual optimization. Hence, a set of swarm-based optimization techniques, namely Grey Wolf (Mirjalili et al., 2014), Whale (Mirjalili and Lewis, 2016), and Particle Swarm (Kennedy and Eberhart, 1995), were employed to assess and optimize the performance of LightGBM and other two ML models. The selection of optimization algorithms is motivated by their proven strengths. Grey Wolf Optimization (GWO) excels in balancing exploration and exploitation, ensuring swift convergence to global optima (Saheed and Misra, 2024). Similarly, the Whale Optimization Algorithm (WOA) offers a high probability of escaping local optima and is less reliant on initial solutions (Gharehchopogh and Gholizadeh, 2019). Finally, Particle Swarm Optimization (PSO) was included in the experiments for its simplicity and its widespread use in the literature, despite being an older algorithm. It has proven effective in enhancing optimization problems (Benbouhenni et al., 2024).

The remainder of the paper is structured as follows: Section 2 encompasses the materials and methods, delineating the processes involved in data gathering and presenting information about the attributes of the dataset. Following this, the section includes an explanation for SMOTE technique (Chawla et al., 2002), succeeded by an introduction to LightGBM, RF, ET, and the associated optimization techniques. Section 3 provides details regarding the experimental framework and the metrics observed throughout the experiments. Section 4 presents the outcomes of the experiments along with their interpretations. Finally, Section 5 concludes the paper by discussing its limitations and suggesting potential avenues for future work.

## 2. Material and Methods

### *2.1. Dataset*

This study employed a recently curated dataset obtained through the Spotify API. The dataset was prepared according to users' preferences and their affinity for songs. The Spotify's API provides various numerical attributes pertaining to a designated song. A brief explanation for each attribute supplied by the Spotify's API is given in Table 2. Also, the distribution of each attribute is given in Figure 1.

**Table 2.** Attributes

| Attribute Name | Explanation | Value |
|---|---|---|
| Acousticness | Confidence level of song's acousticness | Real value between 0-1 |
| Danceability | Whether the song is suitable for dancing | Real value between 0-1 |
| Energy | Energy level of a song | Real value between 0-1 |
| Instrumentalness | Whether the song is verbal or not | Real value between 0-1 |
| Liveness | Whether the song has audience or not | Real value between 0-1 |
| Loudness | Loudness of the song in decibels | Real value between -60 – 0 Db |
| Duration | Duration of the song in milliseconds | - |
| Mode | Whether the song's melodic content is major or minor | Either 0 or 1 |
| Speechiness | Whether words are present in the song | Real value between 0-1 |
| Tempo | Tempo level of the song | Real value |
| Valence | Level of positiveness of the song | Real value between 0-1 |

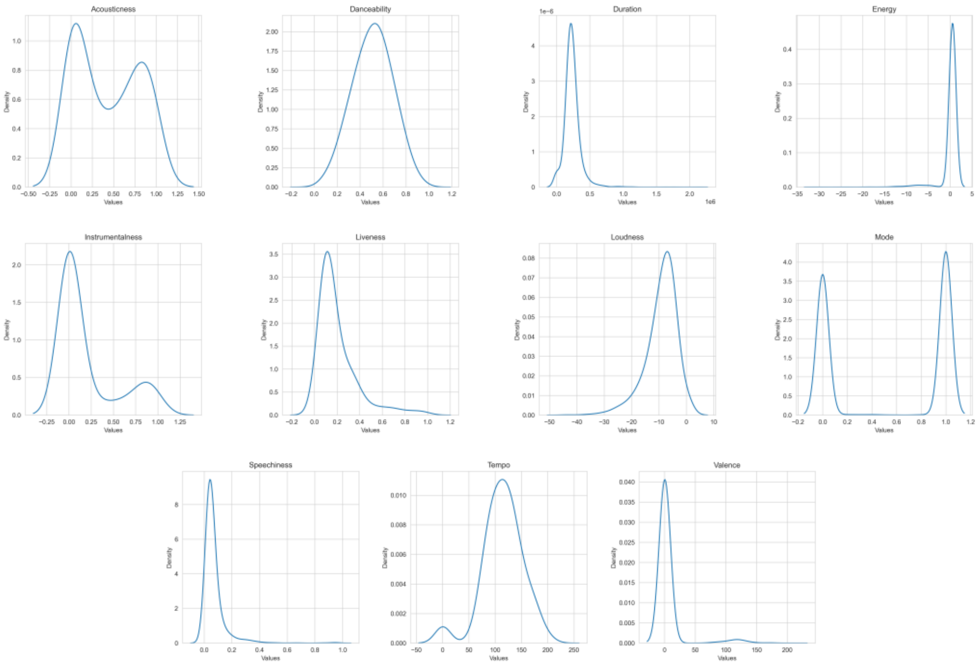



**Figure 1.** Distribution of the Attributes

A label was added to each song's attribute list. It signifies the user's inclination towards the song. A value of 0 (zero) denotes that the song was not find favored by the user, whereas a value of 1 (one) signifies the converse. Consequently, the resulting dataset transforms the music recommendation system into a classical binary classification problem in ML. The cumulative count of songs in the dataset, following this procedure, amounted to 5462. Given the potential variance in users' preferences, the dataset exhibits notable imbalances with respect to labels. The label distribution of the dataset is given in Figure 2.

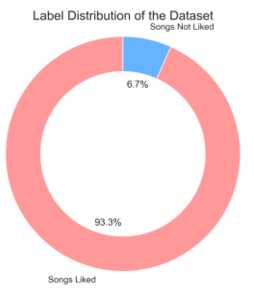



**Figure 2.** Label Distribution of the Dataset

As illustrated in Figure 2, the dataset exhibits a high level of imbalance, a characteristic commonly encountered in the field of AI. In the pursuit of establishing robust ML frameworks, it is imperative to ensure dataset balance. This requirement emerges from the necessity for ML models to have equal exposure to each label category. Ultimately, achieving a balanced dataset in real-world scenarios is not always feasible, given the labor-intensive nature of the process. For this reason, this resource-intensive process may be facilitated through the generation of synthetic data based on the observed data. One of the predominant methodologies utilized for this purpose is referred to as SMOTE, and it was incorporated in this study. The subsequent section imparts succinct information on the SMOTE algorithm for the benefit of the reader.

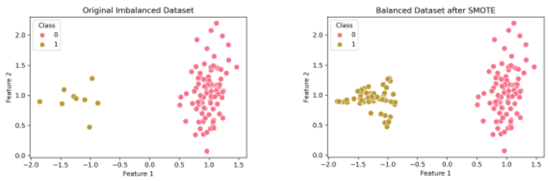
### *2.2. SMOTE*

The majority of ML datasets available on the internet are generally well-balanced and meticulously curated. Consequently, these curated datasets can be utilized without the necessity for further modification. However, real-life curated datasets do not necessarily exhibit this property, and generally present issues related to data imbalance. For this reason, it is imperative to address this imbalance by either collecting additional data or employing synthetic data generation techniques to balance the distribution of data labels. One of the techniques that are used in this area is SMOTE. The overall algorithm is formulated through the process of interpolation, involving diverse instances from the minority class located within a predefined neighborhood (Fernández et al., 2018).

The mathematical formula for SMOTE is given in Equation 1.

$$x_{new} = x_i + \lambda(x_j - x_i) \tag{1}$$

where $x_{new}$ is the new generated sample, $x_i$ is an instance from the minority class, $x_j$ is randomly selected neighbor of $x_i$ from $k$ nearest neighbor. Finally, $\lambda$ is a random number between 0 and 1. A toy, graphical example of SMOTE is given in Figure 3.




(a) (b)

**Figure 3.** A Graphical Example of SMOTE. (a) Imbalanced Data. (b) Balanced Data by SMOTE

In the context of the ML framework, the study leveraged the capabilities of the LightGBM, ET and RF algorithms for classifying song labels. The next two subsections provide a concise overview of the algorithms.

### *2.3. LightGBM*

LightGBM, introduced by Microsoft (Ke et al., 2017), is an acronym for Light Gradient Boosting Machine. It constitutes an ensemble-based method commonly applied to ML problems, including regression and classification. One of the major advantages of the LightGBM algorithm is that it employs a histogram-based learning methodology for the discretization of features. This entails binning continuous feature values into discrete bins, thereby mitigating the computational burden associated with determining the optimal split during the tree growth process. Moreover, it incorporates regularization terms within its objective function to mitigate the risk of overfitting. The inclusion of regularization aids in managing the model's complexity, fostering enhanced generalization performance on previously unseen data.

### *2.4 Random Forest and Extra Tree Algorithms*

The Random Forest (RF) algorithm, categorized as an ensemble method, employs an internal ensemble of multiple trees, aggregates their predictions to improve accuracy. Each constituent tree within the ensemble selectively samples from the original dataset. While there are similarities between Extra Tree (ET) and RF, a significant divergence is apparent in their construction methodologies. Unlike RF, which utilizes diverse sub-samples during model construction, ET employs the entire dataset. Furthermore, ET introduces randomness in node splitting, while RF selects optimal features for this purpose. The next section briefly provides information about the rationale behind optimization and introduces these optimization algorithms.

### *2.5. Optimization*

Optimization techniques play a pivotal role in enhancing the performance of ML algorithms by fine-tuning their parameters to attain optimal results. These techniques are designed to navigate the extensive parameter space effectively, seeking the combination that minimizes a predefined objective function. The iterative process involves systematically exploring the parameter space to identify values that optimize the desired outcome. Generally, optimization algorithms are model-free, meaning that they can be applied to any kind of problem, as long as a suitable objective function is provided. Following of this section, optimization algorithms employed in this study are briefly introduced.

### *2.6 Grey Wolf Optimization Algorithm*

Derived from the hunting and social dynamics of grey wolves in nature, the Grey Wolf Optimizer (GWO) algorithm has emerged as a metaheuristic optimization approach renowned to solve the optimization problems (Mirjalili et al., 2014).

Capitalizing on the principle that nature serves as the ultimate optimizer, the GWO algorithm incorporates the roles of alpha, beta, and delta wolves, symbolizing the leadership within a wolf pack. This utilization aims to steer the search for optimal solutions.

The algorithm's prowess in exploration and exploitation is orchestrated through collaborative efforts among the wolves. During the exploration phase, the alpha wolf takes the lead, while the beta wolf concentrates on exploitation. The delta wolf plays a crucial role in introducing a balance between these two essential aspects.

### *2.7 Whale Optimization Algorithm*

Similar to GWO, Whale Optimization Algorithm (WOA) mimics the behavior of humpback whales (Mirjalili and Lewis, 2016). It is based on cooperative hunting strategies employed by the whales. While GWO has the concept of alpha, beta and delta, whales have the ability to encircling, spiral updating, and prey search mechanisms. Also, those can be defined as exploration phase, encircling phase, and exploitation phase. The encircling phase identifies a candidate solution towards the optimal solution (prey), guiding the search process. During the exploration phase, each whale's position undergoes random changes, fostering a diverse exploration of the solution space. In contrast, the exploitation phase represents a more systematic approach than the exploration phase. Here, the algorithm employs a strategy known as the Bubble-Net, enabling the systematic exploitation of the local area surrounding the optimal solution (prey).

### *2.8 Particle Swarm Optimization Algorithm*

Final optimization algorithm employed in this study is Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995). This time, the optimization algorithm leverages the behavior of a flock of birds that moves as a group. Each bird (solution) in a flock employs three different properties, a position, a best position, and finally a velocity that determines how much change in each direction (problem dimension) must be done.

All these three optimization algorithms can be classified as swarm-based algorithms where swarm means in this context is searching a solution space collaboratively. Easy adaption of these algorithms makes them ideal candidates for the optimization of ML algorithms employed in this study. The next section contains the detailed explanation of the experiments conducted.

## 3. Experiments

The aforementioned imbalance in the dataset was addressed by employing the SMOTE prior to the commencement of experiments. Given the single parameter involved in SMOTE, we opted for a neighbor's range of 4. The balanced dataset using SMOTE is given in Figure 4.

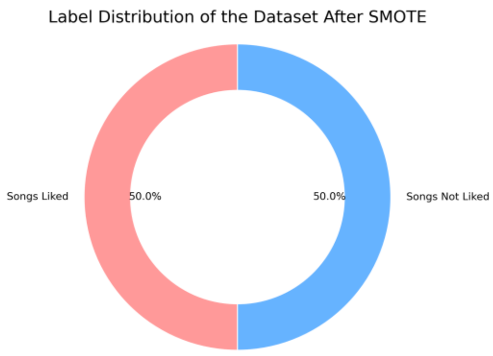



**Figure 4.** Label Distribution of the Dataset after SMOTE

Before proceeding with the experiments, the dataset needs to be partitioned into training and test sets. In this stage K-Fold cross validation was employed and the hyperparameter K was selected as 10 (Kohavi, 1995). The overall dataset was divided into 10 equal size folds. Then each ML model was evaluated 10 times, with each fold serving as the testing set once and the remaining folds used for training.

To optimize the LightGBM and other algorithms efficiently, the choice of a suitable fitness function is pivotal. In our framework, the most suitable criterion for this purpose is to enhance the algorithms based on their accuracy. All three optimization algorithms were configured to maximize the accuracy of the ML models. Since K-Fold cross validation technique was applied during training, average accuracy on the test portions of the 10-Folds utilized as the performance metric. The fitness function used in the experiment is given in Equation 2.

$$F_{accuracy} = \sum_{a=1}^{10} (\gamma(y_a = \hat{y}_a))\,/10 \tag{2}$$

where $y_a$ is the Kth fold (test fold) of the dataset and $\gamma$ can be defined in Equation 3.

$$\gamma\,(y = \hat{y}) = \begin{cases} 1 \; if \; y \; = \hat{y} \\ 0 \; if \; y \, ! = \hat{y} \end{cases} \tag{3}$$

Additionally, the training process included the evaluation of other performance metrics, namely the F1 score, Recall, and Precision, for which the formulations are provided in Equation 4, Equation 5, and Equation 6, respectively.

$$F1\ Score = \frac{2\ x\ Precision\ x\ Recall}{Precision + Recall} \tag{4}$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{5}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{6}$$

Due to the relatively high number of parameters in the ML models used in this study, we chose to optimize only the parameters that are common to each model. The selected parameters and their lower and upper bounds used in the experiments are given in Table 3.

**Table 3.** Parameters and Their Search Space

| Parameter | Lower – Upper Bounds |
|---|---|
| Number of Estimators (NE) | 100 - 500 |
| Max Depth (MD) | 8 - 16 |
| Max Leaf Nodes (MLN) | 2 - 1024 |

To objectively evaluate the performance of all optimizers, each optimizer was executed for 20 generations, with each generation comprising 10 individuals. All features in the dataset were normalized to speed up to convergence. The experiments were conducted in Python (Version 3.10.13) programming language.

## 4. Results

We commenced the presentation of our results by directly showcasing the accuracy and optimized parameters achieved by all optimizers. The best results and the optimized values of the parameters are provided in Table 4 and Table 5 respectively.

**Table 4.** Results of the ML Models

| | GWO | | | | PSO | | | | WOA | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Accuracy** | **F1 Score** | **Precision** | **Recall** | **Accuracy** | **F1 Score** | **Precision** | **Recall** | **Accuracy** | **F1 Score** | **Precision** | **Recall** |
| **ET** | 92.35% | 92.51% | 90.75% | 94.70% | 91.33% | 91.57% | 89.13% | 94.28% | 92.42% | 92.57% | 90.71% | 94.56% |
| **RF** | 94.43% | 94.41% | 94.61% | 94.35% | 94.47% | 94.48% | 94.54% | 94.64% | 94.69% | 94.69% | 94.59% | 94.96% |
| **LightGBM** | 96.60% | 96.58% | 97.35% | 96.01% | 96.65% | 96.63% | 97.34% | 96.04% | 96.61% | 96.58% | 97.34% | 95.93% |

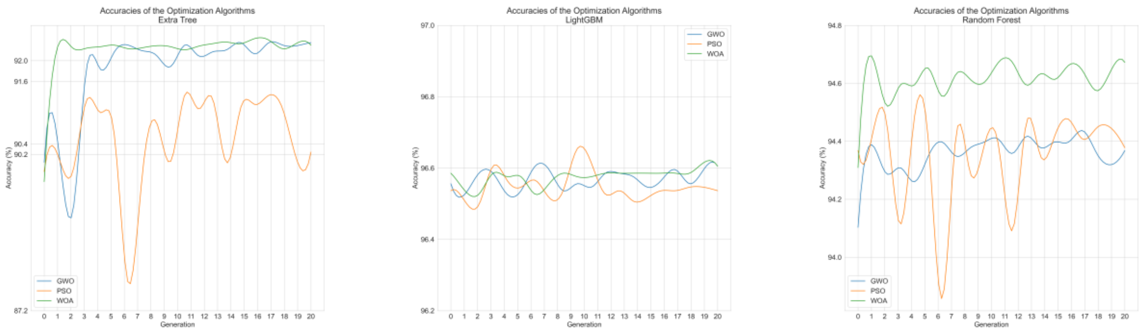
**Table 5.** Optimized Parameters

| | GWO | | | PSO | | | WOA | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Number of Estimators** | **Max Depth** | **Max Leaf Nodes** | **Number of Estimators** | **Max Depth** | **Max Leaf Nodes** | **Number of Estimators** | **Max Depth** | **Max Leaf Nodes** |
| ET | 198 | 16 | 1022 | 474 | 16 | 920 | 500 | 16 | 1024 |
| RF | 477 | 16 | 704 | 269 | 16 | 710 | 500 | 16 | 1024 |
| LightGBM | 314 | 15 | 682 | 256 | 15 | 244 | 416 | 14 | 978 |

As shown in Table 4, LightGBM outperforms the other two models, with ET exhibiting the lowest performance, achieving an accuracy of 92.42% at best. Similarly, the RF algorithm achieved an accuracy of 94.69% at best. However, when considering all three optimizers, LightGBM demonstrated superior performance, achieving an accuracy of 96.65% at best. These observations hold true for other performance metrics as well, indicating that LightGBM consistently outperformed the other two ML models. For each ML model, their metric performances across optimizers were similar. Similar observations can be made regarding the optimized parameter values for ET and RF. All three optimizers converged to a maximum depth of 16.

However, there were notable differences in the optimized number of estimators for GWO and PSO, while WOA converged to the same parameter value for both ET and RF. The convergence to similar parameter values can be attributed to the limited number of parameters used for optimization. However, the parameter values for LightGBM differed across all three optimizers. One important observation is that each optimizer localized the parameters of the ML models to different regions, although the results, such as accuracy, did not vary greatly within each optimizer.

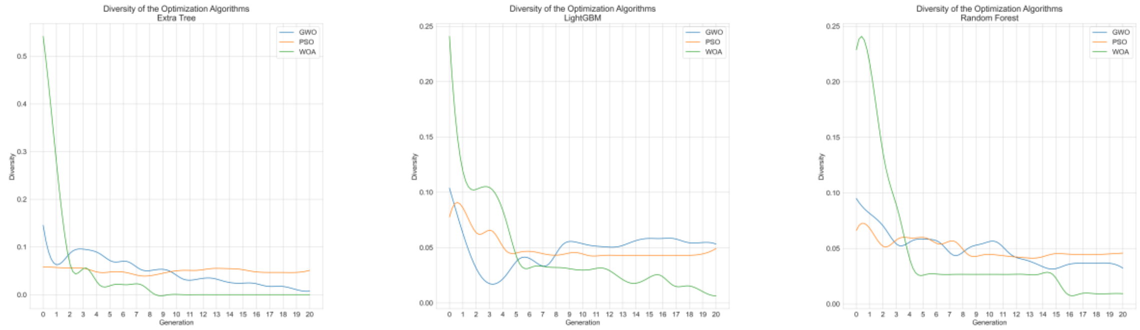
To analyze the performance of the optimizers in depth, a thorough analysis was conducted. Figure 5 shows the highest accuracy attained by each model across all generations.




(a) (b) (c)

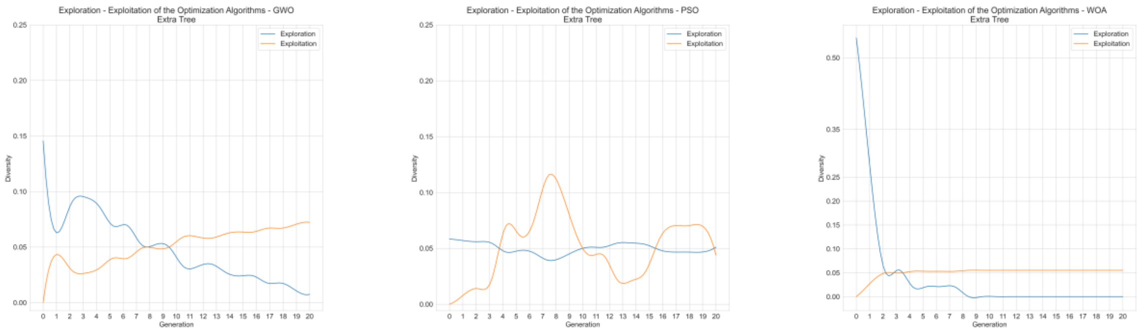
**Figure 5.** Accuracy Graph of the Optimization Algorithms. (a) Extra Tree (b) LightGBM (c) Random Forest

Figure 5 illustrates the performance of each optimizer for each model. The first notable difference is observed in the accuracy of the LightGBM algorithm optimized by all three optimizers, which appeared to oscillate at certain intervals during the generations. When considering the other two ML models, especially PSO stands out as the divergent algorithm among all three optimizers. The other two optimizers (GWO and WOA) exhibited more consistent performance across generations when considering ET and RF. Although all optimizers seemed to converge, GWO and WOA achieved this convergent earlier than PSO. For ET, WOA and GWO seemed to localize after 5$^{th}$ generation. However, PSO did not seem to localize as WOA and GWO for ET. Divergence of PSO can be seen more clearly in the optimization of RF. GWO and WOA had less divergence after 5$^{th}$ generation, whereas divergence seemed to be much less for PSO after 13$^{th}$ generation. One general deduction is that GWO and WOA were, in general, more stable across generation while PSO oscillated greatly between generations. To enhance the understanding the behaviors of optimizers, Figure 6 presents the diversity graphs of all optimizers for each ML model. Finally, graphs of exploration – exploitation for each optimizer for each ML model are given in Figure 7.

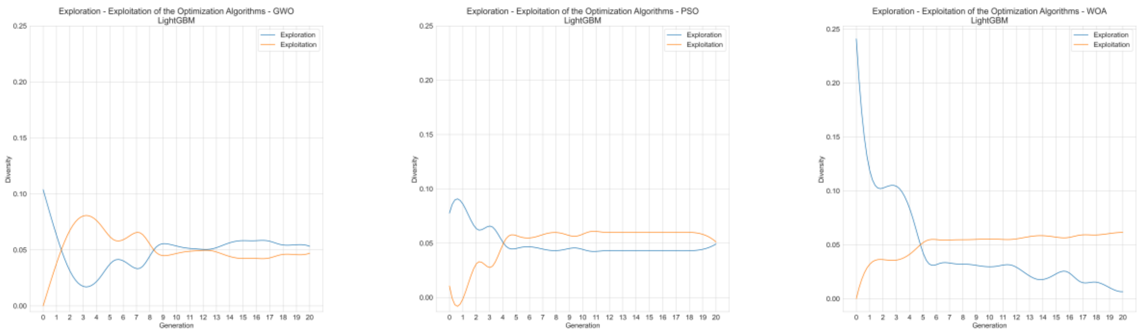



(a) (b) (c)

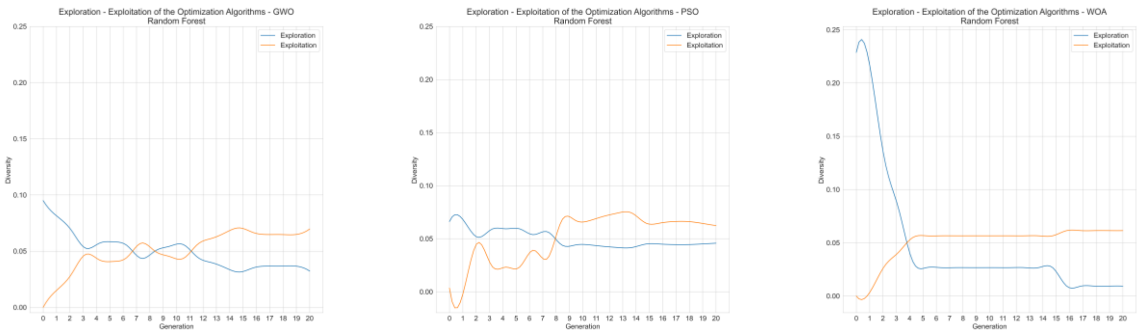
**Figure 6.** Diversity of the Optimization Algorithms for Each Model. (a) Extra Tree (b) LightGBM (c) Random Forest




(a) (b) (c)




(d) (e) (f)




(g) (h) (i)

**Figure 7**. Exploration - Exploitation of the Optimization Algorithms for Each ML Model. (a) ET - GWO (b) ET – PSO (c) ET – WOA (d) LightGBM – GWO (e) LightGBM – PSO (f) LightGBM – WOA (g) RF – GWO (h) RF – PSO (i) RF - WOA

As expected, the diversity of all three optimization algorithms decreased as each generation produced more stable outcomes. Initially, WOA exhibited high divergence, but as generations evolved, this divergence decreased. Divergence of PSO did not change as highly as ET and RF. This property could explain the oscillation characteristic that was given in Figure 5. For the LightGBM algorithm, there was a significant drop observed for PSO. This behavior could also explain the slight divergence in LightGBM accuracies as depicted in Figure 5.

When Figure 7 is examined, as expected, all optimization algorithms began with a high exploration rate. However, WOA seemed to exhibit the highest exploration rate at the beginning of the optimization process. As generations evolved, the exploration rate dropped and the exploitation rate seemed to rise, indicating that all optimizers were not discovering new locations but rather concentrating more on localization. In general, all optimizers exhibited similar behaviors to each other. However, PSO in ET optimization showed inconsistent behaviors in exploration – exploitation phase (Figure 7 (b)). This behavior is also consistent with the diversity of PSO given in Figure 6 (a).

As indicated by the results and more in-depth analyses, all optimization algorithms exhibited nearly similar behaviors across all ML models. However, the LightGBM algorithm was optimized more steadily compared to the other two ML models. ET and RF nearly stabilized at the same values for their parameters. However, LightGBM localized entirely different parameter spaces. Since LightGBM has different parameters than ET and RF, its parameters that were not optimized in this study had profound effect on the results. Lastly, a concise comparison is presented in Table 6 to situate this study within the context of existing literature on music recommendation.

**Table 6**. Comparison of the Studies

| Study | Dataset | ML Model | Performance |
|---|---|---|---|
| (Yuwono et al., 2023) | Spotify | SVM | 80% |
| (Wen et al., 2024) | GTZAN | CNN | 91.4% |
| (Soekarta et al., 2023) | GTZAN | CNN | 72% |
| (HIZLISOY et al., 2023) | GTZAN | Extra Tree | 92.3% |
| This study | Spotify | LightGBM | 96.65% |

Although this study focused on binary classification, it can be easily enhanced for multiclass classification by incorporating genre label to each song. From Table 6, two important insights emerge: first, the LightGBM algorithm could be considered as a viable choice for song recommendation. Secondly, the attributes extracted from the Spotify API demonstrate their utility in AI systems within the music industry, yielding highly competitive results.

## 5. Conclusion

Music recommendation is a challenging process that various independent variables must be considered which may influence individuals' preferences. Solely depending on the genres that an individual listened may not be enough to produce reliable music recommendation systems. Also, the proposed system must be optimized and achieve the best result possible. The current study introduces a novel dataset derived exclusively from individuals' preferences for music pieces, curated utilizing the Spotify API. Recognizing the common occurrence of imbalances in real-life datasets, the SMOTE technique was employed to address and rectify the dataset's imbalance. Furthermore, the study utilizes the LightGBM algorithm to categorize music pieces based on users' preferences, distinguishing between liked and not liked songs. Moreover, the LightGBM algorithm was compared with two other ML models similar to LightGBM, namely ET and RF. Finally, all ML models were optimized using three robust optimization algorithms, namely GWO, WOA, and PSO. A thorough analysis was conducted. The results revealed that the LightGBM exhibited superior performance among these ML models.

Tested optimization algorithms comprised entirely of swarm-based optimizers. A potential avenue for future research involves comparing these optimizers with a Deep Neural Networks (DNNs) optimized using algorithms such as Stochastic Gradient Descent (SGD) (Bottou, 2012) or Adam (Kingma and Ba, 2017).

## References


Bartolomeo, P., 2022. Can music restore brain connectivity in post-stroke cognitive deficits? Med. Hypotheses 159, 110761.

Benbouhenni, H., Hamza, G., Oproescu, M., Bizon, N., Thounthong, P., Colak, I., 2024. Application of fractional-order synergetic-proportional integral controller based on PSO algorithm to improve the output power of the wind turbine power system. Sci. Rep. 14, 609. https://doi.org/10.1038/s41598-024-51156-x

Bottou, L., 2012. Stochastic Gradient Descent Tricks, in: Montavon, G., Orr, G.B., Müller, K.-R. (Eds.), Neural Networks: Tricks of the Trade, Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 421–436. https://doi.org/10.1007/978-3-642-35289-8_25

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. 16, 321–357.

Farajzadeh, N., Sadeghzadeh, N., Hashemzadeh, M., 2023. PMG-Net: Persian music genre classification using deep neural networks. Entertain. Comput. 44, 100518.

Fernández, A., Garcia, S., Herrera, F., Chawla, N.V., 2018. SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. J. Artif. Intell. Res. 61, 863–905.

Gentili, G., Simonutti, L., Struppa, D.C., 2023. Music: numbers in motion.

Geurts, P., Ernst, D., Wehenkel, L., 2006. Extremely randomized trees. Mach. Learn. 63, 3–42. https://doi.org/10.1007/s10994-006-6226-1

Gharehchopogh, F.S., Gholizadeh, H., 2019. A comprehensive survey: Whale Optimization Algorithm and its applications. Swarm Evol. Comput. 48, 1–24.

Hawkins, V., 2022. Music-Color Synesthesia: A Historical and Scientific Overview. Aisthesis Honors Stud. J. 13.

Hızlısoy, S., Arslan, R.S., Çolakoğlu, E., 2023. Music Genre Recognition Based on Hybrid Feature Vector with Machine Learning Methods. Çukurova Üniversitesi Mühendis. Fakültesi Derg. 38, 739–750.

Ho, T.K., 1995. Random decision forests, in: Proceedings of 3rd International Conference on Document Analysis and Recognition. IEEE, pp. 278–282.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y., 2017. Lightgbm: A highly efficient gradient boosting decision tree. Adv. Neural Inf. Process. Syst. 30.

Kennedy, J., Eberhart, R., 1995. Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks. IEEE, pp. 1942–1948.

Kingma, D.P., Ba, J., 2017. Adam: A Method for Stochastic Optimization. https://doi.org/10.48550/arXiv.1412.6980

Kohavi, R., 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Ijcai. Montreal, Canada, pp. 1137–1145.

Li, Z., Liu, F., Yang, W., Peng, S., Zhou, J., 2021. A survey of convolutional neural networks: analysis, applications, and prospects. IEEE Trans. Neural Netw. Learn. Syst.

Liu, Z., Xu, W., Zhang, W., Jiang, Q., 2023. An emotion-based personalized music recommendation framework for emotion improvement. Inf. Process. Manag. 60, 103256.

Logan, B., 2000. Mel frequency cepstral coefficients for music modeling., in: Ismir. Plymouth, MA, p. 11.

Loukas, S., Lordier, L., Meskaldji, D., Filippa, M., Sa De Almeida, J., Van De Ville, D., Hüppi, P.S., 2022. Musical memories in newborns: A resting-state functional connectivity study. Hum. Brain Mapp. 43, 647–664. https://doi.org/10.1002/hbm.25677

Mirjalili, S., Lewis, A., 2016. The whale optimization algorithm. Adv. Eng. Softw. 95, 51–67.

Mirjalili, S., Mirjalili, S.M., Lewis, A., 2014. Grey wolf optimizer. Adv. Eng. Softw. 69, 46–61.

Noble, W.S., 2006. What is a support vector machine? Nat. Biotechnol. 24, 1565–1567.

Păvăloaia, V.-D., Necula, S.-C., 2023. Artificial intelligence as a disruptive technology—a systematic literature review. Electronics 12, 1102.

Prabhakar, S.K., Lee, S.-W., 2023. Holistic approaches to music genre classification using efficient transfer and deep learning techniques. Expert Syst. Appl. 211, 118636.

Risse, M., 2023. Political Theory of the Digital Age: Where Artificial Intelligence Might Take Us. Cambridge University Press.

Saheed, Y.K., Misra, S., 2024. A voting gray wolf optimizer-based ensemble learning models for intrusion detection in the Internet of Things. Int. J. Inf. Secur. https://doi.org/10.1007/s10207-023-00803-x

Singh, Y., Biswas, A., 2023. Lightweight convolutional neural network architecture design for music genre classification using evolutionary stochastic hyperparameter selection. Expert Syst. 40, e13241. https://doi.org/10.1111/exsy.13241

Soekarta, R., Aras, S., Aswad, A.N., 2023. Hyperparameter Optimization of CNN Classifier for Music Genre Classification. J. RESTI Rekayasa Sist. Dan Teknol. Inf. 7, 1205–1210.

Wen, Z., Chen, A., Zhou, G., Yi, J., Peng, W., 2024. Parallel attention of representation global time–frequency correlation for music genre classification. Multimed. Tools Appl. 83, 10211–10231.

Wijaya, N.N., Muslikh, A.R., 2024. Music-Genre Classification using Bidirectional Long Short-Term Memory and Mel-Frequency Cepstral Coefficients. J. Comput. Theor. Appl. 2, 13–26.

Yılmaz P., Akçakaya, Ş., Özkaya, Ş.D., Çetin, A., 2022. Machine Learning Based Music Genre Classification and Recommendation System. El-Cezeri 9, 1560–1571.

Yuwono, A., Tjiandra, C.A., Owen, C., Manuaba, I.B.K., 2023. Music Genre Classification Using Support Vector Machine Techniques, in: 2023 International Conference on Information Management and Technology (ICIMTech). IEEE, pp. 511–516.

Zhao, J., Zhao, M., Yang, X., Li, X., Chen, Z., 2023. Music Style Recognition Method Based on Computer-Aided Technology for Internet of Things.