Gazi Üniversitesi
Fen Bilimleri Dergisi
PART C: TASARIM VE TEKNOLOJİ

Gazi University
Journal of Science
PART C: DESIGN AND TECHNOLOGY

GU J Sci, Part C, 13(1): 367-381 (2025)

Enhancing Multi-Disease Prediction with Machine Learning: A Comparative Analysis and Hyperparameter Optimization Approach

Mariam KİLİ BECHIR[1], Ferhat ATASOY[2*]

[1]*Karabuk University, The Institute of Graduate Programs, Department of Computer Engineering, Karabuk, Turkey*

[2]*Karabuk University, Faculty of Engineering, Department of Computer Engineering, Karabuk, Turkey*

Article Info

Research article
Received: 26/05/2024
Revision: 05/02/2025
Accepted: 09/02/2025

Keywords

Machine Learning
Supervised Learning
Multi-Disease Prediction
Hyperparameter Optimization
User-Friendly Application

Makale Bilgisi

Araştırma makalesi
Başvuru: 26/05/2024
Düzeltme: 05/02/2025
Kabul: 09/02/2025

Anahtar Kelimeler

Makine Öğrenmesi
Denetimli Öğrenme
Çoklu-hastalık Tahmini
Hiperparamtre Optimizasyonu
Kullanıcı-dostu Uygulama

Graphical/Tabular Abstract (Grafik Özet)

This study evaluates the performance of several supervised ML models with hyperparameter optimization for predicting multiple diseases such as diabetes, heart disease, Parkinson's disease, and breast cancer. The results show that the performance increase provided by pre-processing and HPO is unfortunately not directly applicable to all datasets. / Bu çalışma, diyabet, kalp hastalığı, Parkinson hastalığı ve meme kanseri gibi birden fazla hastalığı tahmin etmek için hiperparametre optimizasyonlu çeşitli denetimli ML modellerinin performansını değerlendirmektedir. Sonuçlar, ön işleme ve HPO tarafından sağlanan performans artışının ne yazık ki tüm veri kümelerine doğrudan uygulanamayacağını göstermektedir.

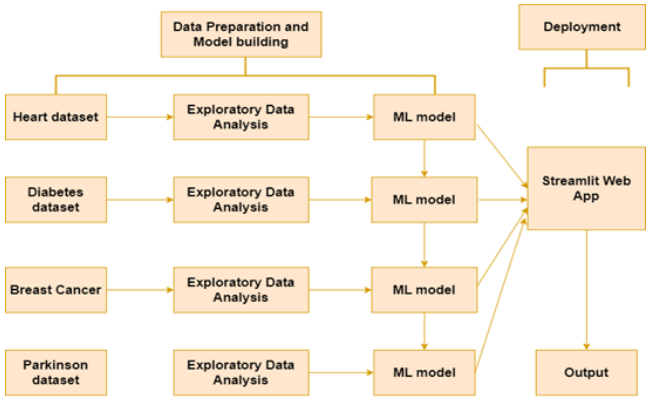

***Figure A**: Architecture design of the system / **Şekil A:**Sistemin mimari tasarımı*

Highlights (Önemli noktalar)

- *User-friendly web Application for multi-disease prediction / Çoklu hastalık tahmini için kullanıcı dostu web uygulaması*
- *Importance of HPO and preprocessing in machine learning / Makine öğrenmesinde HPO ve önişlemenin önemi*
- *Explaratory data analysis / Keşifçi veri analizi*

***Aim (Amaç):** The aim of this study is to investigate the use of ML methods in the diagnosis of diabetes, heart disease, breast cancer and Parkinson's disease and the effect of HPO on model performance. / Bu çalışmanın amacı diyabet, kalp hastalığı, meme kanseri ve Parkinson hastalığının teşhisinde MÖ yöntemlerinin kullanımı ve HPO'nun model performansına etkisinin araştırılmasıdır.*

***Originality (Özgünlük):** In this study, the success of exploratory data analysis and preprocessing and hyperparameter optimization on different health data sets was evaluated. It was revealed that there is no successful method that can be applied to all data sets. / Bu çalışmada keşifçi veri analizi ve önişleme ile hiperparametre optimizasyonunun farklı sağlık veri setlerindeki başarısı değerlendirilmiştir. Tüm veri setlerine uygulanabilir, başarılı bir yöntem olmadığı ortaya koyulmuştur.*

***Results (Bulgular):** Each dataset should be specifically processed with its data types, missing and outlier values and optimized to produce a robust model. / Her veri seti veri türleri, eksik ve aykırı değerleri ile özel olarak işlenmeli gürbüz model ortaya koymak için optimize edilmelidir.*

***Conclusion (Sonuç):** Four different diseases were classified with six different ML methods, and the performances before and after HPO were compared and it was found that the optimized methods were more successful. / Dört farklı hastalık altı farklı ML yöntemiyle sınıflandırılmış, HPO öncesi ve sonrası performansları kıyaslanarak optimize yöntemlerin daha başarılı olduğu bulunmuştur.*

*Corresponding author, e-mail: ferhatatasoy@karabuk.edu.tr
DOI: 10.29109/gujsc.1489959

GU J Sci, Part C, 13(1): 367-381 (2025)

Gazi Üniversitesi
Fen Bilimleri Dergisi
PART C: TASARIM VE TEKNOLOJİ

Gazi University
Journal of Science
PART C: DESIGN AND TECHNOLOGY

http://dergipark.gov.tr/gujsc

Enhancing Multi-Disease Prediction with Machine Learning: A Comparative Analysis and Hyperparameter Optimization Approach

Mariam KİLİ BECHIR[1], Ferhat ATASOY[2*]

[1]*Karabuk University, The Institute of Graduate Programs, Department of Computer Engineering, Karabuk, Turkey*

[2]*Karabuk University, Faculty of Engineering, Department of Computer Engineering, Karabuk, Turkey*

Article Info

Research article
Received: 26/05/2024
Revision: 05/02/2025
Accepted: 09/02/2025

Keywords

Machine Learning
Supervised Learning
Multi-Disease Prediction
Hyperparameter Optimization
User-Friendly Application

Abstract

This study evaluates the performance of supervised machine learning (ML) models with hyperparameter optimization (HPO) for predicting multiple diseases, including diabetes, heart disease, Parkinson's disease, and breast cancer. Six algorithms—Logistic Regression (LR), Gradient Boosting (GB), k-Nearest Neighbors (K-NN), Extreme Gradient Boosting (XGB), Support Vector Machines (SVM), and Random Forests (RF) were trained and tested on specific disease datasets. Model performance was assessed using accuracy, precision, recall, and F1-score, with GridSearch-based HPO applied to enhance predictive accuracy. Significant improvements were observed across datasets. For heart disease, accuracy increased from 85.43% to 99.49% after HPO, with similar gains in other metrics. In diabetes prediction, KNN accuracy improved from 82.10% to 86.32%, while precision and F1-score also rose. Breast cancer models, except XGBoost, consistently achieved over 97.0% accuracy. For Parkinson's disease, SVM achieved 91.17% accuracy, a 4.83% improvement with HPO, with approximately 5% increases across all metrics. All results were validated using 5-fold cross-validation. A user-friendly web application was developed to allow users to select a disease, input relevant data, and receive predictions based on the chosen model. This study highlights the impact of pre-processing and HPO on model performance, addressing computational complexity, demonstrating generalizability in multi-disease prediction, and improving accessibility. However, the results also indicate that the performance gains from pre-processing and HPO are not uniformly applicable across all datasets, providing valuable insights for future research.

Makine Öğrenmesiyle Çoklu Hastalık Tahmininin İyileştirilmesi: Karşılaştırmalı Analiz ve Hiperparametre Optimizasyon Yaklaşımı

Makale Bilgisi

Araştırma makalesi
Başvuru: 26/05/2024
Düzeltme: 05/02/2025
Kabul: 09/02/2025

Anahtar Kelimeler

Makine Öğrenmesi
Denetimli Öğrenme
Çoklu-hastalık Tahmini
Hiperparamtre Optimizasyonu
Kullanıcı-dostu Uygulama

Öz

Bu çalışma, diyabet, kalp hastalığı, Parkinson hastalığı ve meme kanseri dahil olmak üzere birden fazla hastalığı tahmin etmek için hiperparametre optimizasyonu (HPO) ile denetlenen makine öğrenimi (MÖ) modellerinin performansını değerlendirir. Altı algoritma - Lojistik Regresyon (LR), Gradient Boosting (GB), k-En Yakın Komşular (K-NN), Extreme Gradient Boosting (XGB), Support Vector Machines (SVM) ve Random Forests (RF) - belirli hastalık veri kümeleri üzerinde eğitildi ve test edildi. Model performansı, tahmin doğruluğunu artırmak için GridSearch tabanlı HPO uygulanarak doğruluk, kesinlik, geri çağırma ve F1 puanı kullanılarak değerlendirildi. Veri kümeleri arasında önemli iyileştirmeler gözlemlendi. Kalp hastalığı için doğruluk, HPO'dan sonra %85,43'ten %99,49'a yükselirken, diğer metriklerde de benzer kazanımlar elde edildi. Diyabet tahmininde, KNN doğruluğu %82,10'dan %86,32'ye yükselirken, kesinlik ve F1 puanı da arttı. XGBoost hariç meme kanseri modelleri sürekli olarak %97,0'ın üzerinde doğruluk elde etti. Parkinson hastalığı için SVM, HPO ile %4,83'lük bir iyileştirme olan %91,17 doğruluk elde etti ve tüm metriklerde yaklaşık %5 artış oldu. Tüm sonuçlar 5 katlı çapraz doğrulama kullanılarak doğrulandı. Kullanıcıların bir hastalık seçmesine, ilgili verileri girmesine ve seçilen modele dayalı tahminler almasına olanak tanıyan kullanıcı dostu bir web uygulaması geliştirildi. Bu çalışma, ön işleme ve HPO'nun model performansı üzerindeki etkisini, hesaplama karmaşıklığını ele almayı, çoklu hastalık tahmininde genelleştirilebilirliği göstermeyi ve erişilebilirliği iyileştirmeyi vurgulamaktadır. Ancak sonuçlar ayrıca ön işleme ve HPO'dan elde edilen performans kazanımlarının tüm veri kümelerinde tekdüze olarak uygulanabilir olmadığını ve gelecekteki araştırmalar için değerli içgörüler sağladığını göstermektedir.

*Corresponding author, e-mail: ferhatatasoy@karabuk.edu.tr
DOI: 10.29109/gujsc.1489959

1. INTRODUCTION (GİRİŞ)

In a world where the diversity of complex diseases is increasing day by day, accurate diagnoses have underscored the need for advanced healthcare technologies. This is vital both for patients to access treatment quickly and for healthcare professionals to reduce their workload. In addition to these, the increasing risk of people making mistakes under work intensity and intense stress increases the importance of using Artificial Intelligence (AI) in health.

Medical data analysis is increasingly used in modern healthcare to improve diagnosis, treatment, and prognosis processes. ML, as a subfield of AI, has the potential to be an effective tool for various fields [1–4]. Unlike conventional methods, ML leverages patterns in data for early detection and risk prediction. Previous studies have shown the success of ML in diagnosing diseases such as Alzheimer's, diabetes, heart disease, Parkinson's, and various cancers [5–14]. ML algorithms offer powerful tools to extract meaningful information from such data. However, the performance of these algorithms is highly dependent on the correct tuning of hyperparameters [15]. Hyperparameter optimization is the process of finding the most appropriate values of the parameters determined by the user during the training process of the model and guiding the learning process of the model. This process is of critical importance especially in complex and sensitive datasets such as medical data [16,17].

The application of ML improves the diagnostic process by facilitating early detection, precise disease classification, and personalized treatment recommendations, thereby significantly improving patient care and prognosis. The difficulties inherent in medical data, such as high dimensionality, unbalanced class distributions, and missing data, make hyperparameter optimization even more complex [18]. Therefore, in addition to traditional optimization methods, it is recommended to use more advanced methods such as Bayesian optimization, grid search, and random search [19]. In addition, measures to improve model performance on medical data (e.g., data preprocessing, feature selection, and class balancing techniques for imbalanced data sets) need to be carefully examined [20]. Recent studies have emphasized ensemble models and HPO to achieve improved diagnostic accuracy and address overfitting concerns [21].

In this study, we investigate the diagnosis of diseases such as diabetes, heart disease, breast cancer and Parkinson's disease using ML methods and the effect of HPO on models' performances. The performance of six models for each disease on four different datasets is analyzed and presented. In addition, a synthesis of existing studies in the literature will be presented and suggestions for future research in this area will be made. While prior studies have focused on individual disease prediction, such as [22] achieving respectively 85.04% and 84.4% accuracy with LR and RF for heart disease, our work uniquely evaluates multiple diseases, including heart disease, diabetes, breast cancer, and Parkinson's, under a unified framework. Additionally, the integration of HPO across all models usually enhances their robustness compared to studies like [23], which primarily relied on default configurations.

With the web-based user-friendly application developed in the study, it has become possible to predict the probability of an individual developing a disease. This application has been developed to bridge the gap between technical innovation and practical usability, providing accessible predictions for clinical and general users.

2. RELATED WORKS (İLGİLİ ÇALIŞMALAR)

In our study, we considered literature in two dimensions: 1- The effect of machine learning methods and hyperparameter optimization, 2- Disease classification using machine learning methods.

2.1. Hyperparameter Optimization (Hiperparametre Optimizasyonu)

Hyperparameter optimization is an important process to increase the performance of a model. At the end of this process, the aim is to find the most appropriate values of the parameters of the algorithms used. Generally, each algorithm has its own parameters, and adjusting these parameters appropriately affects the results according to the performance metrics used to measure the generalization ability and success of the model [15,19,24–29].

In this study, Logistic Regression (LR), Random Forest (RF), Support Vector Machines (SVM), K-Nearest Neighbor (KNN), Extreme Gradient Boosting (XGBoost) and Gradient Boosting (GB) algorithms were used.

Although LR is generally considered as a simple model, adjusting the regularization term (C) prevents the model from overfitting or undefitting, contributing to the increase in generalization ability. In addition, the selection of different optimization algorithms called solvers is another parameter that affects the training time and accuracy [30–35].

In the random forest algorithm, adjusting hyperparameters such as the number of trees (n_estimators), maximum depth (max_depth) and minimum sample number (min_samples_split) can significantly increase performance. Generally, more trees provide increased performance, but it increases computational cost. While it becomes possible to solve more complex problems with maximum depth, it causes overfitting and the training process to take longer. The minimum sample number determines the number of samples required to split a node, and as this value increases, the generalization ability of the model can be increased. Model accuracy can be increased with the most appropriate choices. In addition, by preventing overfitting or underfitting situations, the generalization ability and stability of the model can be increased and provided. In addition, similar or better results can be obtained by reducing computational complexity. As a result of the research, the hyperparameter optimization process is seen as an important step in random forest applications [24,36–40].

In SVM, optimization of hyperparameters such as kernel function, C parameter and gamma parameter can increase the classification success of the model. Kernel function determines how the data will be separated. Linear, polynomial and Radial Basis Function (RBF) are commonly used. C parameter controls the error tolerance. While low values tolerate more errors, high values create more complex models with less errors. Gamma parameter determines how effective the data will be when RBF kernel is used. While low gamma values provide a wider domain, high values create a narrower domain. Research shows that correct adjustment of these parameters has a significant effect on the overall performance of the model. For example, the overall success of the model can be increased with appropriate C and gamma values. The selection of the kernel function and optimization of hyperparameters can directly affect the shape and location of the decision boundaries and therefore the classification performance. In addition, it is possible to improve the computational efficiency and the training process [41–43].

There are several critical hyperparameters of the KNN algorithm. The number of neighbors (k) can improve the accuracy of the model, especially depending on the size of the dataset. While low k values increase the sensitivity of the model to noise, high k values can increase the generalization ability of the model. The distance criterion used to determine the neighbors can affect the classification results. In addition, the weighting method is used to determine the effects of the neighbors. HPO provides accuracy, minimizing over- or under-fitting, and computational efficiency as in other algorithms. The number of neighbors and the distance criterion are important in determining the decision boundaries, and their total effect can directly affect the classification performance [15,24,44–46].

XGBoost and GB algorithms are algorithms that use decision trees. XGBoost is an optimized and faster version of the traditional GB algorithm. Both add new trees that correct the weak points of the current model to reduce the error at each iteration. XGboost offers more advantages over GB with parallel processing, pruning mechanism used to prune excess trees, special target functions and a wide range of loss functions. The prominent hyperparameters to be optimized are the number of trees to be created (n_estimators), learning rate (eta), maximum depth (max_depth), sample rate (subsample). As the number of ‘n_estimators’ increases, the results generally improve, while the risk of overfitting also increases. ‘eta’ controls the contribution of each tree. Lower ‘eta’ values can help increase generalization ability by providing slower and better learning. ‘subsample’ determines the ratio of samples to be used for each tree. Lower values can reduce overfitting. HPO can provide performance increase as in other methods [19,27,37,47–50].

2.2. Disease Classification Using ML Methods
(MÖ Yöntemleri Kullanarak Hastalık Sınıflandırma)

Recent advances in machine learning (ML) have reshaped disease prediction, yet critical challenges in generalizability, interpretability, and clinical integration remain unresolved. Previous studies have highlighted the effectiveness of ML in disease prediction. For example, in study [21] various ML algorithms were used to train models for predicting 141 diseases across different medical specialties, including diabetes, bronchial asthma, and Covid-19. The research introduced a valuable dataset for healthcare-oriented ML research and demonstrated the potential of these algorithms for multi-disease prediction with high accuracies. By creating a

comprehensive dataset and applying diverse ML algorithms, including SVM, Naïve Bayes, and Random Forest, the study addressed the challenges of multi-disease prediction. The achieved accuracy of 99.33% suggests the potential of ML for this task. However, more investigation is necessary to ensure the generalizability and robustness of such models across different healthcare settings. While this study focuses on the use of large-scale symptom datasets, their reliance on association rules and homogeneous data sources raises concerns about robustness in real-world clinical settings, a limitation that our work addresses through rigorous hyperparameter optimization (HPO) and validation across heterogeneous datasets.

According to [51] an improved healthcare efficiency by predicting ICD codes for chronic diseases, achieving a prediction accuracy between 80-90% across 11 diseases, with even higher accuracy for cancer and stroke. This work demonstrated ML's potential to speed up diagnosis through automated code prediction. Similarly, [52] evaluated whether ML can increase the accuracy of cardiovascular risk estimation using regular clinical data from a prospective cohort of 378,256 patients registered dataset. Four ML algorithms (RF, LR, GBM, ANN) were compared to an existing algorithm based on American College of Cardiology guidelines. The findings showed that ML algorithms significantly improved cardiovascular risk estimation. In particular, the neural networks algorithm showed the best improvement in accuracy, with a 3.6% increase over traditional methods, demonstrating ML's ability to outperform existing risk assessment models. However, early ML studies focused narrowly on single diseases or homogeneous datasets, overlooking the need for generalizable frameworks applicable to diverse clinical contexts.

Other research focused on comparing specific ML models for chronic diseases. Study [53] compared LR with various ML models for predicting chronic diseases like diabetes, cardiovascular disease, hypertension, and chronic kidney disease. The results demonstrated that LR achieved competitive performance, particularly for diseases with well-established risk factors. In study [54], the researchers aimed to develop a framework that could accurately distinguish between Parkinson patients and healthy individuals at an early stage. They employed various ML algorithms to analyze data and identify patterns indicative of the disease. Preprocessing, standardization, and ensemble techniques were utilized to enhance model performance. The resulting DL models demonstrated promising outcomes in discriminating Parkinson patients from healthy individuals, as evaluated using accuracy, precision, sensitivity, F1 score, specificity, and area under the ROC curve. It is emphasized that since the study used a limited dataset, the real performance of the DL method can be demonstrated with larger datasets.

Hyperparameter optimization (HPO) is another key process for improving ML model efficacy. In [55], the hyperparameters of the LR, KNN, SVM, RF, and Decision Tree (DT) classification models were tuned using the grid search method, resulting in an accuracy range of 81.97% to 90.16% for LR, KNN, DT, and SVM, and an accuracy range of 85.25% to 91.80% for RF. Similarly, in [23], the efficacy of heart disease prediction was evaluated using HPO techniques. The success rate, which typically ranges between 80 and 90% in the literature, was found to be 97.52% in the Cleveland dataset, particularly when the RF model was employed. This indicates a notable improvement in performance using HPO.

In addition, [56] proposed a study that concerns cardiac disease diagnosis. The study emphasizes the use of pre-processing and feature engineering techniques before employing the ML algorithms. These techniques aim to clean and refine the data by handling outliers, missing values, potentially irrelevant features and improve the performance of ML models.

In [57], authors proposed a DL model for multi-classification of infectious diseases utilizing unstructured electronic medical records to aid in clinical decision-making regarding infectious diseases. This research highlights the capability of deep learning to manage complex, unstructured data such as electronic medical records, enhancing the classification of diseases. The presented model achieved a significantly higher accuracy (99.44%) compared to traditional ML algorithms like XGBoost (96.19%), DT (90.13%), Bayesian methods (85.19%), and LR (91.26%). The model demonstrates the potential of DL in analyzing complex, unstructured data like EMRs, offering a more comprehensive approach to disease classification compared to traditional methods. However, existing HPO studies focus disproportionately on single-disease applications or computationally heavy DL models, neglecting lightweight, interpretable frameworks for multi-disease systems.

3. MATERIALS AND METHODS (MATERYAL VE METOD)

3.1. Datasets (Veri Setleri)

To examine the effectiveness of ML with and without HPO in the healthcare field, we have applied different ML methodologies on 4 different medical datasets. These datasets are Diabetes dataset, heart dataset, Breast Cancer dataset and Parkinson dataset. These data are all available on Kaggle data collection platform and lastly accessed on September 6, 2024 for our study.

For the diabetes dataset (https://www.kaggle.com/datasets/mathchi/diabetes-data-set), specific criteria were applied during data selection. This subset focuses on female patients over 20 years old with a specific heritage. It contains 768 entries with eight characteristics, including blood pressure, glucose levels, and a target variable for prediction.

The heart disease dataset (https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset) combines data from multiple sources. This dataset consists of 303 records with 14 characteristics encompassing demographics, medical history, and test results. Additionally, a target variable indicates the presence or absence of heart disease.

The breast cancer dataset (https://www.kaggle.com/datasets/yasserh/breast-cancer-dataset) provides data from 1988 related to breast tissue samples. It includes 569 records, each representing a cell nucleus with 31 characteristics and a target variable indicating malignancy. These characteristics describe various properties extracted from digitized images.

The Parkinson's disease dataset (https://www.kaggle.com/datasets/vikasukani/parkinsons-disease-data-set) contributes significantly to developing ML models for automated disease detection through voice analysis. This dataset comprises information on 195 individuals, aiming to differentiate between those with Parkinson's disease and those who are healthy.

3.2. Exploratory Data Analysis and Variable Analysis (Keşifçi Veri Analizi ve Değiken Analizi)

The target distribution of the diabetes dataset in the first row of Table 1 shows that the person who are not diabetic are more than the person who have diabetes. Around 500 are non-diabetics and 268 people are affected by diabetes.

The target variable distribution in the second row of Table 1 indicates a slightly higher proportion of individuals classified as "affected" compared to those classified as "non-affected". The total number of people affected by heart disease is 526 and the total number of the persons not affected is 499.

The target variable distribution in the third row of Table 1 shows a significantly higher proportion of individuals classified as "affected" compared to those classified as "non-affected". The total number of people affected by heart disease is 147 and the total number of the persons not affected is 48.

Looking at the output of the target variable of the breast cancer dataset int the last row of Table 1, the total number of persons that are Benign (not affected) is 357 and the total number of those who are Malignant (affected by breast cancer) is 212.

Table 1. The target distribution of datasets (Veri setlerinin hedef dağılımları)

	Healthy Number of People	Diseased Number of People
Diabetes	500	268
Heart Disease	499	526
Parkinson	48	147
Breast Cancer	357	212

3.3. Outlier Removal and Standardization (Aykırı Değerlerin Giderilmesi ve Standardizasyon)

Although ML algorithms are powerful, some of their limitations cannot be ignored. Unfortunately, mechanisms that perform exploratory data analysis and train the most optimal model as a result have not yet been developed. The data preprocessing step involved identifying and removing outliers present in each dataset. Outliers are significantly differing from most of the data. Their presence can negatively impact model training and lead to inaccurate predictions. We used Interquartile Range (IRQ) to detect and remove outliers from each dataset.

Then, all features were standardized with the Z-score method known as standardscaler in the Sklearn library. Since these were not always

sufficient, additional operations had to be performed on a dataset-specific basis.

The IQR (Interquartile Range) method is a widely used statistical technique for identifying outliers. This method is based on the distribution of the data set and uses the interquartile range to detect outliers. Here are the steps of the IQR method:

1. Sorting the Data:
   The dataset is sorted from smallest to largest.

2. Calculating the Quartiles:
   Q1 (First Quartile): Represents the lower 25% of the dataset.
   Q3 (Third Quartile): Represents the upper 25% of the dataset.
   Median (Q2): The median value of the data set.

3. IQR Calculation:
   IQR is the difference between Q3 and Q1:

$$IQR = Q3 - Q1 \quad (1)$$

4. Determining Outlier Boundaries:

Lower Boundary:

$$\text{Lower Boundary} = Q1 - 1.5 \times IQR \quad (2)$$

Upper Boundary:

$$\text{Upper Boundary} = Q3 + 1.5 \times IQR \quad (3)$$

5. Detecting Outliers:
   Values smaller than the lower boundary are considered lower outliers.
   Values larger than the upper boundary are considered upper outliers.

The IQR method is a simple and fast method that is based on dataset distribution and objectively determines outliers. It is especially useful for detecting outliers in small and medium-sized datasets. Since it is designed according to the principle that the dataset should be close to a normal distribution, it does not always succeed in providing the same advantage [58].

Following outlier management, we standardized the remaining data. Standardization is a crucial step that scales the features within each dataset to a common range. This method standardizes the data so that the mean is 0 and the standard deviation is 1. This is especially important for improving the performance of machine learning algorithms because it ensures that features at different scales are given equal weight. It speeds up the training process, especially for gradient descent-based algorithms (e.g., SVM, logistic regression, neural networks), and standardized data can be more easily interpreted in statistical analyses [59].

$$z = \frac{(x - \mu)}{\sigma} \quad (4)$$

Here: x: Original value, μ: Mean of the feature, σ: Standard deviation of the feature, z: Standardized value.

3.4. Data Splitting and Balancing (Veri Bölme ve Dengeleme)

k-fold cross-validation evaluates the performance of the model on the entire dataset, providing more reliable results compared to a single training-test split. Since the entire dataset is used for both training and testing, the risk of overfitting the model to the entire dataset is reduced and data waste is prevented. The k value was selected as 5 in the study [60].

SMOTE improves the performance of classification models by increasing the number of samples from the minority class. In addition, since no samples are removed from the dataset (undersampling is not performed), there is no loss of information. Since it creates synthetic samples instead of random oversampling, it reduces the risk of overfitting. Thus, it increases the number of samples from the minority class to solve the class imbalance problem and improves the generalization ability of the model [61].

3.5. Logistic Regression (Lojistik Regresyon)

LR is a supervised learning algorithm that estimates the likelihood of an event (classification) based on a set of input variables (data points). It excels at providing clear explanations for each variable's impact on the predicted classification outcome. This interpretability is valuable in healthcare settings, as it allows us to understand which factors are most influential in predicting each condition.

3.6. k-Nearest Neighbour (k-En Yakın Komşu)

This is a versatile technique used for exploring relationships within data. It operates on the principle of similarity, where a new data point is classified based on the majority class or average value of its closest neighbors. The key aspect of KNN is finding the optimal number of neighbors (k) that minimizes

errors. Selecting the appropriate k value is crucial, as a small k can be sensitive to noise, while a large k might result in an overly smooth decision boundary that misses local patterns.

3.7. Ensemble Techniques (Topluluk Teknikleri)

Ensemble techniques were used to potentially improve the accuracy and robustness of the classifications. Random Forest Classifier combines multiple decision trees, each trained on a random subset of the data, to arrive at a final classification through a majority vote. Extreme Gradient Boosting (XGBoost) creates and combines multiple decision trees, each focusing on correcting the errors of the previous trees, to achieve a more accurate final classification. Gradient Boosting builds an ensemble of decision trees sequentially, where each subsequent tree learns from the errors of the previous trees. This allows the final model to capture complex relationships within the data.

3.8. Support Vector Machine (Destek Vektör Makinesi)

This technique excels in data with a high number of variables. SVM seeks to identify the optimal dividing line that separates different classifications within the data. The positioning of this line prioritizes maximizing the separation between classes.

3.9. Hyperparameter Optimization (Hiperparametre Optimizasyonu)

HPO is a crucial step in ML that involves adjusting the model's internal configurations to optimize its performance. Different hyperparameter settings can significantly impact the model's ability to learn from the data and generalize to unseen data. We used Grid SearchCV which is a common technique that systematically evaluates a predefined set of hyperparameter values to identify the combination that yields the best performance on a validation set [59].

3.10. Model Evaluation (Model Değerlendirmesi)

When using k-fold cross-validation in the scikit-learn library, the performance metrics of the model are calculated for each fold and the final result is usually obtained by averaging these values.

K-Fold Cross-Validation Process:

1. The dataset is divided into k parts (folds).
2. In each iteration, 1 fold is used as the test set and the rest as the training set.
3. The model is trained and tested k times.
4. Metrics (accuracy, precision, recall, F1-score, etc.) are calculated for each fold.
5. The final performance is determined by averaging the results.

Since we used k-fold cross validation, in each iteration performance metrics are calculated k times and final values were calculated by averaging. In addition, we compared results without HPO and with HPO, whole process were operated twice.

The performance metrics of ML models have been calculated and evaluated in terms of accuracy, precision, recall and F1 score with the help of the confusion matrix. Accuracy is the ratio of correctly classified observations to the total number of observations. Precision is performed by taking the ratio of correctly classified positive samples to the total predicted positive samples. The recall is calculated by taking the ratio of truly classified positive samples to all samples in actual class. F1 score is performed by taking the weighted average of precision and recall. The mathematical expressions of accuracy, precision, recall, and F1 score are shown in the equations from 5 to 8 respectively.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$F1 - score = 2\frac{Recall \times Precision}{Recall + Precision} \tag{8}$$

Where, TP, FN, FP, and TN represented True Positive, False Negative, False Positive, and True Negative, respectively.

4. APPLICATION (UYGULAMA)

We used python with pandas to read and manage structured datasets for each disease. Visualization libraries like Matplotlib and Seaborn facilitated data exploration. Each dataset was analyzed independently to understand its characteristics. To address imbalanced datasets, we employed the SMOTE technique. SMOTE generates synthetic data points for the minority class, mitigating bias and improving model performance. A Technique like StandardScaler helped us to standardize the

data, ensuring a mean of 0 and a standard deviation of 1 for all features.

4.1. Data Preprocessing (Veri Önişleme)

We evaluated various ML techniques on the preprocessed data. These included LR, SVM, KNN, RF, XGBoost and GB. HPO was performed using Grid Search method to identify the best configuration for each model, significantly enhancing their performance. k-fold cross-validation strategy ensured good evaluation by providing more reliable performance results.

With the data preprocessing, performing some additional operations for each dataset made significant contributions to the model performance.

Since the values of 'trestbps', 'chol', 'thalach', 'slope', 'thal' used as features in the heart disease dataset are naturally not possible to be 0, the relevant records were cleaned from the dataset. Later, by applying other preprocessing methods, high classification success was achieved by preventing overfitting of the models both without HPO and with HPO.

In the diabetes dataset, it is not possible for the 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI' features to be 0. For this reason, the relevant records were cleaned before other data preprocessing steps. In the experiments conducted with different scenarios, it was observed that the highest performance metrics were obtained in this way.

In the breast cancer dataset, the 'id' column, which has nothing to do with diagnosis, has been deleted. In addition, the features 'radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean', 'concave points_mean', 'radius_se', 'area_se', 'perimeter_worst', 'area_worst', which have high correlations with the target value and each other, have not been included in the training for the purpose of reducing dimensionality. In this way, it has been observed in experimental studies that the models are both faster and give optimum performance.

In the Parkinson's disease dataset, attributes were extracted from voice recordings taken from 31 volunteers. Therefore, outlier detection and removal were applied directly. No other process was done beforehand.

4.2. Model Deployment and Web Application Development (Model Kurulumu ve Web Uygulaması Geliştirme)

The web application was developed using Streamlit, a Python library for building web applications. This application integrates serialized models, enabling users to interact with the system. The architecture of the system is given in Figure 1 and sample interface is given in Figure 2.

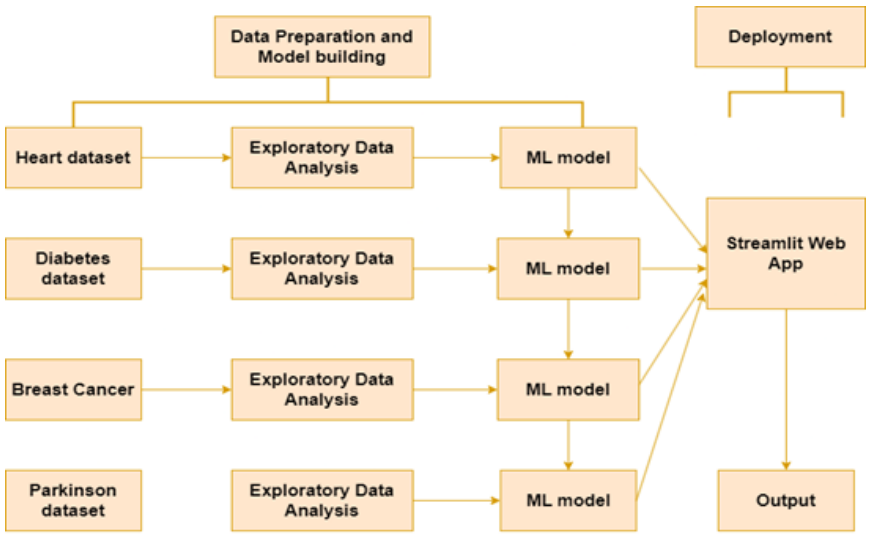

Figure 1. Architecture desing of multi-disease prediction system (Çoklu hastalık tahmin sistemi mimari tasarımı)

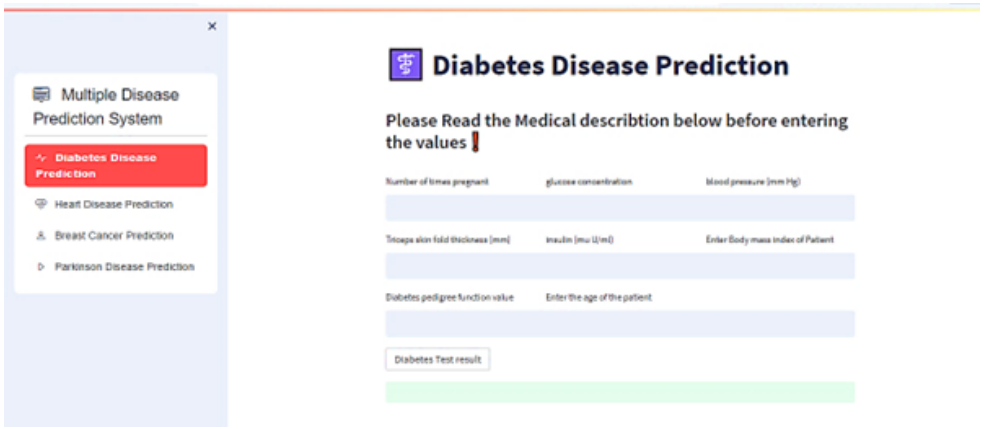

Figure 2. Multi-Disease prediction system user interface (Çoklu hastalık tahmin sistemi kullanıcı arayüzü)

The developed application was tested for usability with simulated clinical and general user inputs. The interface provides a streamlined workflow: disease selection, data input, and instant predictions. Feedback indicated ease of use and potential for integration into clinical workflows, particularly for early screening and risk assessment. Predictions generated by the application matched the model outputs, confirming its reliability. The intuitive interface allows non-technical users to navigate seamlessly, select diseases, input parameters, and interpret results effectively.

5. EXPERIMENTAL RESULTS (Deneysel Sonuçlar)

5.1. Heart Disease (Kalp Hastalığı)

We built and evaluated six ML models for heart disease prediction: LR, RF, SVM, KNN, XGBoost and GB. When we look at the results shown in Table 2, we see that HPO has a significant impact on the performance of the models except RF. KNN showed the most significant improvement (14%), suggesting its initial configuration was suboptimal. This highlights how tuning can help identify better configurations even for simpler models. The chosen hyperparameters (high C value, increased iterations) indicate the model benefits from regularization and more training data. 5-fold cross-validation was used in all training. Cross-validation reduces the chance factor as it allows training and testing with different parts of the dataset. It shows that the model performs well on the entire dataset, not just on a specific subset of the data. However, the increase in KNN brings overfitting to mind. However, since the dataset is balanced before model training, the probability of overfitting decreases due to the difference between training and test scores being around 1%.

Although a small increase is observed in Logistic Regression, it can be expected that the accuracy limits will not change much due to the nature of this model. The best parameters include 'C': 0.1 and 'penalty': 'l1', which shows that the model gives better results with L1 (Lasso) regularization. A striking improvement is seen in KNN. When we look at the best parameters, the settings 'metric': 'manhattan', 'n_neighbors': 11' and 'weights': 'distance' attract attention. Optimizing these parameters seems to have provided the model with significantly better generalization.

Table 2. Comparison of accuracy results on heart disease with and without HPO. (HPO'lu ve HPO'suz kalp hastalıklarında doğruluk sonuçlarının karşılaştırılması.)

Model	Before HPO	After HPO	Best Parameters
LR	83.63%	84.91%	'C': 0.1, 'penalty': 'l1', 'solver': 'liblinear'
KNN	85.43%	99.49%	'metric': 'manhattan', 'n_neighbors': 11, 'weights': 'distance'
RF	99.23%	99.23%	'bootstrap': True, 'max_depth': 70, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200
SVM	92.07%	97.31%	'C': 10, 'kernel': 'rbf'
XG_Boost	99.11%	99.23%	'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 50
GB	97.44%	98.59%	'learning_rate': 0.2, 'max_depth': 5

Interestingly, the accuracy did not change after hyperparameter optimization in RF (99.23%). This shows that the model is already working very close to the optimal values and HPO does not provide any additional improvement. The best parameters determined are depth 70 and number of trees 200, which provides the model to learn strongly. There is a significant increase from 92.07% to 97.31% in SVM. Among the best parameters, 'C': 10 and 'kernel': 'rbf' were selected, meaning the model generalized better with a higher penalty parameter and RBF kernel. The accuracy was already high in the XGBoost and GB algorithms. In both algorithms, the model was optimized with 'learning_rate': 0.2 and 'max_depth' as 7 and 5, respectively.

5.2. Diabetes (Diyabet)

While all models achieved good accuracy after tuning, the results in Table 3 provide insights into model-specific behavior. The KNN model showed the most significant improvement, with test accuracy increasing from 82.1% to 86.3%, a gain of 4.2 percentage points. The Logistic Regression (LR) model saw a slight improvement, whereas the Random Forest (RF) model experienced a minor decline, with accuracy dropping from 83.7% to 83.5%. Surprisingly, the XGBoost model deteriorated, with test accuracy decreasing from 84.2% to 82.3%. The SVM model remained unchanged, and the Gradient Boosting (GB) model showed a slight decline in performance.

Table 3. Comparison of HPO results on diabetes. (HPO'lu ve HPO'suz diyabette doğruluk sonuçlarının karşılaştırılması.)

Model	Before HPO	After HPO	Best Parameters
LR	79.47%	80.26%	'C': 0.1, 'penalty': 'l2', 'solver': 'liblinear'
KNN	82.10%	86.32%	'metric': 'manhattan', 'n_neighbors': 2, 'weights': 'distance'
RF	83.68%	83.53%	'bootstrap': False, 'max_depth': 60, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200
SVM	83.16%	83.16%	'C': 1, 'kernel': 'rbf'
XGBoost	84.21%	82.26%	max_ 'learning_rate': 0.2, 'max_depth': 3, 'n_estimators': 100
GBM	85.26%	85.00%	'learning_rate': 0.1, 'max_depth': 3

Although the LR model improved, its performance gain was not as strong as in other models. The KNN model was successfully optimized, achieving better performance after HPO. However, the choice of a small value (n_neighbors=2) may increase the risk of overfitting, so it should be validated on larger datasets. XGBoost and RF unexpectedly performed worse after HPO. The learning rate was increased in XGBoost (learning_rate=0.2), yet its performance declined. Similarly, the RF model's performance slightly worsened, likely due to excessive model complexity for the dataset. The SVM model remained unchanged, suggesting that its parameters might have already been optimal.

5.3. Breast Cancer (Meme Kanseri)

HPO demonstrated the performance of six models for breast cancer classification (Table 4). HPO provided performance increase in all models except RF model. It is possible to obtain higher scores by expanding the parameter space of RF model. However, since the same parameter pool was used for all models in the study, the result remained low. Although XGBoost and Gradient Boosting work with similar parameters, XGBoost gave slightly better results. As a result, the model that achieved the highest accuracy appears to be KNN with 98.42%.

The best parameters for KNN are seen to be "manhattan" distance metric, 6 neighbors and weighted calculation (distance weighting). The effect of HPO on KNN was quite positive (97.37% → 98.42%). For LR, especially the 'C' value of 10 and the selection of 'liblinear' solver may be effective. Using linear kernel and 'C' = 1 in SVM provided stable development. For XGBoost, 'learning_rate = 0.2', 'max_depth= 3' and 'n_estimators=200' trees provided better results. A slight decrease is seen in the RF model after HPO. This situation can be explained by the fact that too much depth (max_depth = 60) or other parameters used reduce the generalization ability of the model.

Table 4. Comparison of HPO results on diabetes. (HPO'lu ve HPO'suz meme kanseri doğruluk sonuçlarının karşılaştırılması.)

Model	Before HPO	After HPO	Best Parameters
LR	96.49%	97.54%	'C': 10, 'penalty': 'l2', 'solver': 'liblinear'
KNN	97.37%	98.42%	'metric': 'manhattan', 'n_neighbors': 6, 'weights': 'distance'
RF	97.72%	97.19%	'bootstrap': False, 'max_depth': 60, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100
SVM	97.37%	97.54%	'C': 1, 'kernel': 'linear'
XGBoost	96.67%	97.10%	'learning_rate': 0.2, 'max_depth': 3, 'n_estimators': 200
GB	96.84%	97.02%	'learning_rate': 0.2, 'max_depth': 3

A small improvement was seen in the GB model. Similar to the XGBoost model, 'max_depth = 3' and 'learning_rate = 0.2' were used, but XGBoost performed slightly better. This suggests that XGBoost may be a more powerful model compared to GB in terms of optimization.

5.4. Parkinson Disease (Parkinson Hastalığı)

HPO results for Parkinson's disease varied significantly between models (Table 5). The accuracy of SVM after HPO increased by 4.83%. C value (10) and RBF kernel may have created more complex distinction surfaces. The accuracy of KNN increased from 88.46% to 90.48% with Manhattan distance, 'n_neighbors= 2' and weighted distance method. The number of small neighbors may have adapted to the data more precisely.

LR was optimized with 'C = 0.1', 'penalty = l2' parameters and solver method 'lbfgs'. 'max_depth= 70', and 'n_estimators= 1000' are aggressive parameters for RF. Although more trees provide better generalization, it can increase the risk of overfitting. Typical powerful settings such as 'learning_rate = 0.1', 'max_depth = 5', 'n_estimators = 600' are used in the XGboost. There was no change in GB (87.03%); 'learning_rate = 0.1' and 'max_depth = 3' can already be optimal.

Since the whole process is performed with 5-fold cross validation, the results can be considered reliable. Models such as SVM and KNN seem to have a significant increase in test accuracy.

Table 5. Comparison of HPO results on Parkinson. (HPO'lu ve HPO'suz Parkinson doğruluk sonuçlarının karşılaştırılması.)

Model	Accuracy Before HPO	Accuracy After HPO (%)	Best Parameters
LR	81.54%	82.92%	'C': 0.1, 'penalty': 'l2', 'solver': 'lbfgs'
KNN	88.46%	90.48%	'metric': 'manhattan', 'n_neighbors': 2, 'weights': 'distance'
RF	86.96%	87.68%	'bootstrap': True, 'max_depth': 70, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 1000
SVM	86.34%	91.17%	'C': 10, 'kernel': 'rbf'
XGBoost	85.63%	86.32%	'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 600
GB	87.03%	87.03%	'learning_rate': 0.1, 'max_depth': 3

6. RESULTS AND DISCUSSIONS (BULGULAR VE TARTIŞMA)

This study evaluated how hyperparameter optimization affects the performance of different machine learning algorithms for the classification of heart disease, diabetes, breast cancer and Parkinson's disease. The models used were LR, KNN, RF, SVM, XGBoost and Gradient Boosting algorithms. Models were trained using 5-fold cross-validation (k-fold cross-validation) and hyperparameter optimization (HPO) was performed with GridSearch method.

Table 6. Comparison of performance impact of HPO on all datasets. (HPO'nun tüm veri setlerindeki performans karşılaştırması.)

		Accuracy		**Precision**		**Recall**		**F1-Score**	
	Model	Before HPO	After HPO	Before HPO	After HPO	Before HPO	After HPO	Before HPO	After HPO
Heart Disease	LR	83.63	84.91	83.73	85.11	83.63	84.91	83.62	84.89
	KNN	85.43	**99.49**	85.81	**99.50**	85.43	**99.49**	85.38	**99.49**
	RF	**99.23**	99.23	**99.25**	99.24	**99.23**	99.23	**99.23**	99.23
	SVM	92.07	97.32	92.21	97.39	92.07	97.32	92.07	97.32
	XGBoost	99.11	99.23	99.11	99.24	99.11	99.23	99.11	99.23
	GB	97.44	98.59	97.48	98.61	97.44	98.59	97.44	98.59
Diabetes	LR	79.47	80.26	80.16	80.82	79.47	80.26	79.35	80.16
	KNN	82.11	**86.32**	82.70	**87.76**	82.11	**86.32**	82.02	**86.16**
	RF	**83.68**	85.53	**84.36**	86.19	**83.68**	85.53	**83.60**	85.47
	SVM	83.16	83.16	83.47	83.47	83.16	83.16	83.11	83.11
	XGBoost	84.21	85.26	84.67	85.73	84.21	85.26	84.15	85.21
	GB	85.26	85.00	85.82	85.56	85.26	85.00	85.21	84.94
Breast Cancer	LR	97.54	97.37	98.57	98.22	96.49	96.49	97.51	97.35
	KNN	96.32	**97.02**	97.19	**95.92**	95.44	**98.25**	96.26	**97.05**
	RF	**97.54**	97.02	**97.91**	97.25	**97.19**	96.84	**97.54**	97.02
	SVM	96.67	1.83	2.83	3.83	4.83	5.83	6.83	7.83
	XGBoost	96.84	96.84	96.53	96.86	97.19	96.84	96.85	96.84
	GB	96.84	97.02	97.20	97.57	96.49	96.49	96.82	97.00
Parkinson	LR	81.54	82.92	82.27	83.67	81.54	82.92	81.41	82.80
	KNN	88.46	**90.48**	88.63	**90.70**	88.46	**90.48**	88.44	**90.47**
	RF	**86.97**	87.68	**87.26**	87.76	**86.97**	87.68	**86.94**	87.68
	SVM	86.34	91.17	86.64	91.29	86.34	91.17	86.31	91.17
	XGBoost	85.63	86.32	86.45	86.94	85.63	86.32	85.47	86.17
	GB	87.03	87.03	87.57	87.57	87.03	87.03	86.95	86.95

The findings of this study have revealed the effectiveness of disease classification models. However, the generalizability of the study can be increased by using larger and more balanced data sets. In the future, model performance can be further improved by applying different feature selection methods. In addition, deep learning approaches can be effective in classifying complex and observational data, especially Parkinson's. General models can be developed for different diseases using transfer learning methods. Finally, before moving on to clinical applications, the modeled systems should be tested with real hospital data and their real-time performance should be analyzed.

ACKNOWLEDGMENTS (TEŞEKKÜR)

This study is supported by Karabük University Scientific Research Projects Coordinatorship with project number KBÜBAP-24-YL-065.

DECLARATION OF ETHICAL STANDARDS (ETİK STANDARTLARIN BEYANI)

The author of this article declares that the materials and methods they use in their work do not require ethical committee approval and/or legal-specific permission.

Bu makalenin yazarı çalışmalarında kullandıkları materyal ve yöntemlerin etik kurul izni ve/veya yasal-özel bir izin gerektirmediğini beyan ederler.

AUTHORS' CONTRIBUTIONS (YAZARLARIN KATKILARI)

***Mariam KİLİ BECHIR*:** She conducted the experiments and contributed the writing process.

Deneyleri yaptı ve yazım sürecine katkıda bulundu.

***Ferhat ATASOY*:** He analyzed the results and performed the writing process.

Deney sonuçlarını analiz etti ve maklenin yazım işlemini gerçekleştirmiştir.

CONFLICT OF INTEREST (ÇIKAR ÇATIŞMASI)

There is no conflict of interest in this study.

Bu çalışmada herhangi bir çıkar çatışması yoktur.

REFERENCES (KAYNAKLAR)

[1] N. AYDIN ATASOY and F. ÇAKMAK, "Web Tabanlı Sürücü Davranışları Analiz Uygulaması," Gazi Journal of Engineering Sciences, vol. 7, no. 3, pp. 264–276, Dec. 2021, doi: 10.30855/gmbd.2021.03.09.

[2] E. DİKBIYIK, Ö. DEMİR, and B. DOĞAN, "Derin Öğrenme Yöntemleri İle Konuşmadan Duygu Tanıma Üzerine Bir Literatür Araştırması," Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji, vol. 10, no. 4, pp. 765–791, Dec. 2022, doi: 10.29109/gujsc.1111884.

[3] Ö. TONKAL and H. POLAT, "Traffic Classification and Comparative Analysis with Machine Learning Algorithms in Software Defined Networks," Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji, vol. 9, no. 1, pp. 71–83, Mar. 2021, doi: 10.29109/gujsc.869418.

[4] M. B. ER, "Akciğer Seslerinin Derin Öğrenme İle Sınıflandırılması," Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji, vol. 8, no. 4, pp. 830–844, Dec. 2020, doi: 10.29109/gujsc.758325.

[5] R. Alanazi, "Identification and Prediction of Chronic Diseases Using Machine Learning Approach," J Healthc Eng, vol. 2022, 2022, doi: 10.1155/2022/2826127.

[6] I. D. Mienye, Y. Sun, and Z. Wang, "An improved ensemble learning approach for the prediction of heart disease risk," Inform Med Unlocked, vol. 20, Jan. 2020, doi: 10.1016/j.imu.2020.100402.

[7] S. Dhabarde, R. Mahajan, S. Mishra, S. Chaudhari, S. Manelu, and N. S. Shelke, "DISEASE PREDICTION USING MACHINE LEARNING ALGORITHMS", [Online]. Available: www.irjmets.com

[8] S. Vilas and A. M. S. Scholar, "Diseases Prediction Model using Machine Learning Technique", doi: 10.32628/IJSRST.

[9] A. Mujumdar and V. Vaidehi, "Diabetes Prediction using Machine Learning Algorithms," in Procedia Computer Science, Elsevier B.V., 2019, pp. 292–299. doi: 10.1016/j.procs.2020.01.047.

[10] T. H. H. Aldhyani, A. S. Alshebami, and M. Y. Alzahrani, "Soft Clustering for Enhancing the Diagnosis of Chronic Diseases over Machine Learning Algorithms," J Healthc Eng, vol. 2020, 2020, doi: 10.1155/2020/4984967.

[11] S. F. Weng, J. Reps, J. Kai, J. M. Garibaldi, and N. Qureshi, "Can Machine-learning improve cardiovascular risk prediction using routine clinical data?," PLoS One, vol. 12, no. 4, Apr. 2017, doi: 10.1371/JOURNAL.PONE.0174944.

[12] S. Nusinovici et al., "Logistic regression was as good as machine learning for predicting major chronic diseases," J Clin Epidemiol, vol. 122, pp. 56–69, Jun. 2020, doi: 10.1016/J.JCLINEPI.2020.03.002.

[13] J. Al Nahian, A. K. M. Masum, S. Abujar, and M. J. Mia, "Common human diseases prediction using machine learning based on survey data," Bulletin of Electrical Engineering and Informatics, vol. 11, no. 6, pp. 3498–3508, Dec. 2022, doi: 10.11591/eei.v11i6.3405.

[14] N. Aydin Atasoy and A. Faris Abdulla Al Rahhawi, "Examining the classification performance of pre-trained capsule networks on imbalanced bone marrow cell dataset," International Journal of Imaging Systems and Technology , vol. 34, no. 3, May 2024, doi: 10.1002/ima.23067.

[15] J. Bergstra, J. B. Ca, and Y. B. Ca, "Random Search for Hyper-Parameter Optimization Yoshua Bengio," 2012. [Online]. Available: http://scikit-learn.sourceforge.net.

[16] M. Claesen and B. De Moor, "Hyperparameter Search in Machine Learning," Feb. 2015, [Online]. Available: http://arxiv.org/abs/1502.02127

[17] Y. A. Ali, E. M. Awwad, M. Al-Razgan, and A. Maarouf, "Hyperparameter Search for Machine Learning Algorithms for Optimizing the Computational Complexity," Processes, vol. 11, no. 2, Feb. 2023, doi: 10.3390/pr11020349.

[18] A. E. W. Johnson et al., "MIMIC-III, a freely accessible critical care database," Sci Data, vol. 3, May 2016, doi: 10.1038/sdata.2016.35.

[19] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms."

[20] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," 2002.

[21] M. ÇOLAK, T. TÜMER SİVRİ, N. PERVAN AKMAN, A. BERKOL, and Y.

EKİCİ, "Disease prognosis using machine learning algorithms based on new clinical dataset," Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering, vol. 65, no. 1, pp. 52–68, Jun. 2023, doi: 10.33769/aupse.1215962.

[22] F. A. Latifah, I. Slamet, and Sugiyanto, "Comparison of heart disease classification with logistic regression algorithm and random forest algorithm," AIP Conf Proc, vol. 2296, Nov. 2020, doi: 10.1063/5.0030579.

[23] R. Valarmathi and T. Sheela, "Heart disease prediction using hyper parameter optimization (HPO) tuning," Biomed Signal Process Control, vol. 70, p. 103033, Sep. 2021, doi: 10.1016/J.BSPC.2021.103033.

[24] M. Feurer and F. Hutter, "Hyperparameter Optimization," in Automated Machine Learning, 2019, pp. 3–33. doi: 10.1007/978-3-030-05318-5_1.

[25] B. Bischl, J. Richter, J. Bossek, D. Horn, J. Thomas, and M. Lang, "mlrMBO: A Modular Framework for Model-Based Optimization of Expensive Black-Box Functions," Mar. 2017, [Online]. Available: http://arxiv.org/abs/1703.03373

[26] G. Luo, "A review of automatic selection methods for machine learning algorithms and hyper-parameter values," Network Modeling Analysis in Health Informatics and Bioinformatics, vol. 5, no. 1, Dec. 2016, doi: 10.1007/s13721-016-0125-6.

[27] P. Probst and B. Bischl, "Tunability: Importance of Hyperparameters of Machine Learning Algorithms," 2019. [Online]. Available: http://jmlr.org/papers/v20/18-444.html.

[28] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," Neurocomputing, vol. 415, pp. 295–316, Nov. 2020, doi: 10.1016/j.neucom.2020.07.061.

[29] D. J. Hand, "Measuring classifier performance: A coherent alternative to the area under the ROC curve," Mach Learn, vol. 77, no. 1, pp. 103–123, Oct. 2009, doi: 10.1007/s10994-009-5119-5.

[30] T. Hastie, R. Tibshirani, and J. Friedman, "Springer Series in Statistics The Elements of Statistical Learning Data Mining, Inference, and Prediction."

[31] A. Y. Ng, "Feature selection, L 1 vs. L 2 regularization, and rotational invariance," in Twenty-first international conference on Machine learning - ICML '04, New York, New York, USA: ACM Press, 2004, p. 78. doi: 10.1145/1015330.1015435.

[32] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," J R Stat Soc Series B Stat Methodol, vol. 67, no. 2, pp. 301–320, 2005, doi: 10.1111/j.1467-9868.2005.00503.x.

[33] F. Pedregosa FABIANPEDREGOSA et al., "Scikit-learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT ET AL. Matthieu Perrot," 2011. [Online]. Available: http://scikit-learn.sourceforge.net.

[34] G. C. Cawley and N. L. C. Talbot, "On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation," 2010.

[35] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.

[36] L. Breiman, "Random Forests," in Machine Learning, vol. 45, 2001, pp. 5–32. doi: 10.1023/A:1010933404324.

[37] P. Probst, M. N. Wright, and A. L. Boulesteix, "Hyperparameters and tuning strategies for random forest," in Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 9, no. 3, Wiley-Blackwell, 2019. doi: 10.1002/widm.1301.

[38] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas, "How Many Trees in a Random Forest?," 2012, pp. 154–168. doi: 10.1007/978-3-642-31537-4_13.

[39] G. Biau and E. Scornet, "A random forest guided tour," Test, vol. 25, no. 2, pp. 197–227, Jun. 2016, doi: 10.1007/s11749-016-0481-7.

[40] G. Louppe, "Understanding Random Forests: From Theory to Practice," Jul. 2014, [Online]. Available: http://arxiv.org/abs/1407.7502

[41] C. Cortes, V. Vapnik, and L. Saitta, "Support-Vector Networks Editor," Kluwer Academic Publishers, 1995.

[42] B. Schölkopf and A. J. Smola, "Kernel Methods," in Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, 2001, pp. 405–406.

[43] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," 2001. [Online]. Available: www.csie.ntu.edu.tw/

[44] M. M. Deza and E. Deza, Encyclopedia of distances. Springer Berlin Heidelberg, 2009. doi: 10.1007/978-3-642-00234-2.

[45] S. A. Dudani, "The Distance-Weighted k-Nearest-Neighbor Rule," IEEE Trans Syst Man Cybern, vol. SMC-6, no. 4, pp. 325–327, 1976, doi: 10.1109/TSMC.1976.5408784.

[46] R. J. Samworth, "Optimal weighted nearest neighbour classifiers," Ann Stat, vol. 40, no. 5, pp. 2733–2763, Oct. 2012, doi: 10.1214/12-AOS1049.

[47] J. H. Friedman, "Stochastic gradient boosting," 2002. [Online]. Available: www.elsevier.com/locate/csda

[48] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[49] T. Chen and T. He, "xgboost: eXtreme Gradient Boosting," in R Package, 2024.

[50] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," Front Neurorobot, vol. 7, no. DEC, 2013, doi: 10.3389/fnbot.2013.00021.

[51] P. N. Astya, Galgotias University. School of Computing Science and Engineering, Institute of Electrical and Electronics Engineers. Uttar Pradesh Section, Institute of Electrical and Electronics Engineers. Uttar Pradesh Section. SP/C Joint Chapter, and Institute of Electrical and Electronics Engineers, Proceeding, International Conference on Computing, Communication and Automation (ICCCA 2016) : 29-30 April, 2016.

[52] S. F. Weng, J. Reps, J. Kai, J. M. Garibaldi, and N. Qureshi, "Can Machine-learning improve cardiovascular risk prediction using routine clinical data?," PLoS One, vol. 12, no. 4, Apr. 2017, doi: 10.1371/JOURNAL.PONE.0174944.

[53] S. Nusinovici et al., "Logistic regression was as good as machine learning for predicting major chronic diseases," J Clin Epidemiol, vol. 122, pp. 56–69, Jun. 2020, doi: 10.1016/J.JCLINEPI.2020.03.002.

[54] W. Wang, J. Lee, F. Harrou, and Y. Sun, "Early Detection of Parkinson's Disease Using Deep Learning and Machine Learning," IEEE Access, vol. 8, pp. 147635–147646, 2020, doi: 10.1109/ACCESS.2020.3016062.

[55] E. Kabir Hashi and M. Shahid Uz Zaman, "Developing a Hyperparameter Tuning Based Machine Learning Approach of Heart Disease Prediction," Journal of Applied Science & Process Engineering, vol. 7, no. 2, 2020.

[56] D. Hamid, S. S. Ullah, J. Iqbal, S. Hussain, C. A. U. Hassan, and F. Umar, "A Machine Learning in Binary and Multiclassification Results on Imbalanced Heart Disease Data Stream," J Sens, vol. 2022, 2022, doi: 10.1155/2022/8400622.

[57] M. Wang, Z. Wei, M. Jia, L. Chen, and H. Ji, "Deep learning model for multi-classification of infectious diseases from unstructured electronic medical records," BMC Med Inform Decis Mak, vol. 22, no. 1, Dec. 2022, doi: 10.1186/s12911-022-01776-y.

[58] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata, "Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median," J Exp Soc Psychol, vol. 49, no. 4, pp. 764–766, Jul. 2013, doi: 10.1016/j.jesp.2013.03.013.

[59] A. Geron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2nd ed. O'Reilly Media, Inc., 2019.

[60] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-Validation," Encyclopedia of Database Systems, pp. 532–538, 2009, doi: 10.1007/978-0-387-39940-9_565.

[61] "View of SMOTE: Synthetic Minority Over-sampling Technique." Accessed: Feb. 05, 2025. [Online]. Available: https://www.jair.org/index.php/jair/article/view/10302/24590